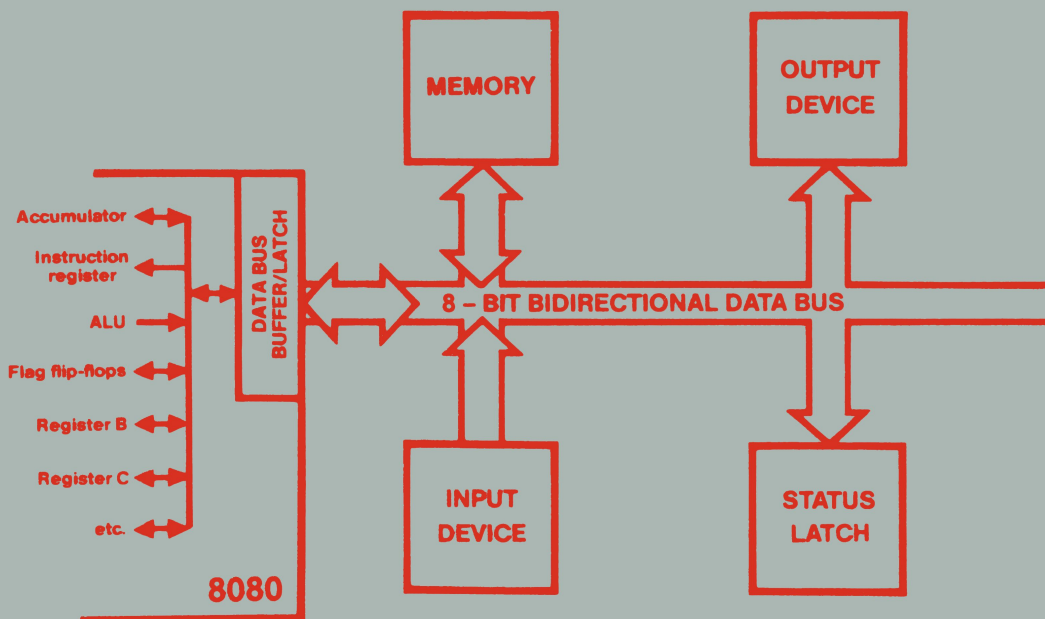
il BUGBOOK® V

ESPERIMENTI INTRODUTTIVI ALL'ELETTRONICA DIGITALE, ALLA PROGRAMMAZIONE E ALL'INTERFACCIAMENTO DEL MICROCOMPUTER 8080A

EDIZIONE ITALIANA

JACKSON ITALIANA EDITRICE

MEMORY

OUTPUT DEVICE

Accumulator

Instruction register

ALU

Flag flip-flops

Register B

Register C

etc.

DATA BUS BUFFER/LATCH

8 – BIT BIDIRECTIONAL DATA BUS

INPUT DEVICE

STATUS LATCH

8080

il
BUGBOOK V°

ESPERIMENTI INTRODUTTIVI ALL'ELETTRONICA
DIGITALE, ALLA PROGRAMMAZIONE
E ALL'INTERFACCIAMENTO DEL MICROCOMPUTER 8080A

DAVID G. LARSEN
Department of Chemistry

PETER R. RONY
Department of Chemical Engineering

Virginia Polytechnic Institute & State University
Blacksburg, Virginia 24061

e

JONATHAN A. TITUS
Tychon, Inc.
Blacksburg, Virginia 24060

Versione italiana

ALDO CAVALCOLI
Mipro s.r.l.
Via Carducci, 15
20123 Milano

VALERIO SCIBILIA
Mipro s.r.l.
Via Carducci, 15
20123 Milano

JACKSON
ITALIANA
EDITRICE
P.le Massari, 22
20125 Milano

INTRODUZIONE ALL'EDIZIONE ITALIANA

La serie dei Bugbook ha rappresentato negli Stati Uniti una pietra miliare nella divulgazione e nell'insegnamento dell'elettronica digitale e delle tecniche di utilizzo dei microprocessori.

I testi sono sorti da esperienze didattiche degli autori, che svolgono attività di ricercatori e docenti presso il Virginia Polytechnic Institute and State University.

Questa esperienza scolastica ad alto livello è una delle caratteristiche più interessanti della serie dei Bugbook, che possono anche essere utilizzati come validi manuali di autoistruzione.

Nella preparazione dell'edizione italiana, abbiamo cercato il più possibile di mantenere lo stile colloquiale degli autori, che del resto dal punto di vista umano sono molto estroversi ed entusiasti della loro attività di ricercatori.

Abbiamo mantenuto molti termini tecnici inglesi sia per il diffuso utilizzo che se ne fa nella corrente terminologia elettronica, sia per familiarizzare il lettore con un certo linguaggio, che sempre più è presente nella letteratura tecnica.

Le apparecchiature descritte nei volumi e utilizzate negli esperimenti hanno avuto origine dall'attività di ricerca degli autori, secondo precise esigenze didattiche; queste apparecchiature sono prodotte dalla E-L Instruments.

Un particolare ringraziamento va a quanti hanno collaborato alla realizzazione della versione italiana dei Bugbook, l'Ing. Ettore Valsecchi, la Sig.na Daniela Fornari, la Sig. ra Rosaria Lucano, il Sig. Marcello Longhini, la Sig.ra Francesca Di Fiore ed il Sig. Giampietro Zanga ideatore della versione italiana dei Bugbook.

Un ringraziamento va anche alle organizzazioni che hanno reso possibile e agevolato il nostro lavoro:

MICROLEM S.a.S. - Via C. Monteverdi, 5 - 20131 Milano - distributore per l'Italia dell'apparecchiatura della E-L Instruments.

MIPRO S.r.l. - Via Carducci, 15 - 20123 Milano - società di consulenza minicomputers-microprocessor.

ELETTRONICA OGGI - Piazzale Massari, 22 - 20125 Milano - rivista mensile di elettronica, microelettronica e automazione professionale.

Aldo Cavalcoli
Valerio Scibilia

SOMMARIO

CAPITOLO 4 - IL MICROCOMPUTER MMD-1

CAPITOLO 5 - ALCUNI SEMPLICI PROGRAMMI PER IL MICROCOMPUTER 8080

CAPITOLO 6 - REGISTRI E ISTRUZIONI RELATIVE AI REGISTRI

CAPITOLO 7 - PORTE LOGICHE E TABELLE DELLA VERITA'

CAPITOLO 8 - ISTRUZIONI LOGICHE

CAPITOLO 9 - INTRODUZIONE AL BREADBOARDING

CAPITOLO 10 - CIRCUITI INTEGRATI

CAPITOLO 11 - FLIP-FLOP E LATCH

CAPITOLO 12 - DECODIFICATORI

CAPITOLO 13 - CONTATORI

CAPITOLO 14 - GATING DI SEGNALI DIGITALI

CAPITOLO 15 - MULTIVIBRATORI MONOSTABILI E ASTABILI

APPENDICI

PREFAZIONE

Benvenuti nella nuova rivoluzione elettronica. In 10 anni la tecnologia dei circuiti elettronici ha trasformato i circuiti integrati digitali da costosi componenti per semplici funzioni logiche, a componenti ad alta complessività, contenenti fino a 10.000 transistor. Da questa rivoluzione è nato il computer-su-singolo-chip! Questo nuovo componente contiene tutto quello che è richiesto da un computer digitale (unità centrale, memoria a lettura/scrittura, memoria ROM, circuiteria di interfaccia). Fra pochi anni sarete in grado di acquistare una manciata di questi componenti a prezzi impensabili. Si ritiene che intorno al 1980 esisterà circa un milardo di microcomputer in circolazione: per avere un'idea del significato esplosivo di questa cifra, basta pensare che attualmente in America vi sono solo 250.000 tra minicomputer e grandi sistemi di elaborazione dati. Una rivoluzione nel campo dei computer? Sicuramente.

Nell'istruzione poi, pensiamo che la nuova rivoluzione elettronica creerà importanti cambiamenti ed opportunità:

- Un numero sempre maggiore di studenti dovrà imparare l'elettronica digitale ed in particolare l'utilizzo dei microcomputer; studenti che diventeranno ingegneri, chimici, biologi, fisici, tecnici agrari, biochimici, psicologi sperimentali.
- I corsi teorici sull'algebra Booleana, sulle mappe di Karnaugh, ed altri argomenti equivalenti, perderanno di importanza per tutti quegli studenti che sono interessati alla tecnologia digitale.
- Gli studenti che si occupano di Scienza dei Calcolatori si troveranno sempre più nella necessità di apprendere nozioni di elettronica digitale: molti studenti avranno il loro proprio microcomputer.
- Centinaia di microcomputer saranno disponibili nell'ambito delle Università; forse migliaia.
- I corsi di telecomunicazioni e controlli digitali cresceranno in importanza.

A fronte di questi cambiamenti, una cosa resterà essenzialmente invariata: il tempo che gli studenti passano a scuola. Gli insegnanti dovranno affrontare il problema di armonizzare gli argomenti sopra indicati con i corsi esistenti, per un inevitabile aggiornamento; sarà anche necessaria la creazione di nuovi corsi e la fusione di altri già in programma, per fornire contenuti didattici validi per l'evoluzione tecnologica ed industriale in atto.

I BUGBOOKS costituiscono una serie di volumi sull'elettronica digitale, sull'interfacciamento dei microcomputer e sulla programmazione dei microcomputer, si tratta di un tentativo di integrazione dei tre argomenti fondamentali succitati in un corso unificato. Questi volumi sono orientati verso esperimenti di laboratorio, che consideriamo la via migliore per esprimere l'eccezionalità e l'importanza della nuova rivoluzione

elettronica. Ai tre argomenti indicati viene data la stessa importanza: imparerete come si programma un microcomputer, come lo si interfaccia verso dispositivi esterni e come i dispositivi esterni operano da un punto di vista digitale. Saranno illustrati importanti concetti di elettronica digitale sia da un punto di vista circuitale, collegando opportuni circuiti integrati, sia da un punto di vista software, realizzando programmi per microcomputer.

Per il lettore di questi volumi, non è necessaria una particolare precedente esperienza in elettronica digitale. Come prima cosa saranno trattati i microcomputer ed i circuiti integrati come *moduli funzionali*. Da questa esposizione verranno apprese in modo graduale le caratteristiche operative di base. Non discuteremo come questi dispositivi sono stati costruiti, in quanto la tecnologia è alquanto sofisticata e si modifica rapidamente.

I Bugbooks sono testi "laboratory oriented", nell'ambito di una serie di libri caratterizzati da un approccio all'elettronica digitale a nostro avviso differente. Piuttosto che iniziare, come è abitudine, con esperimenti sui *componenti elettronici*, come *resistori, condensatori, diodi, transistor,* con questi volumi vi introduciamo direttamente ai concetti relativi ai *circuiti integrati,* i cosiddetti "bugs" dei nostri Bugbooks. Vi forniamo poi immediatamente i concetti di: *switch logici, indicatori a LED, generatori di impulsi, display;* vi mostriamo come utilizzare queste funzioni ausiliarie e vi sottoponiamo esperimenti relativi ai collegamenti tra i circuiti integrati e questi dispositivi.

Una volta in possesso dei concetti base dell'elettronica digitale e delle modalità di collegamento dei vari circuiti, vi introduciamo all'utilizzo di chip e sistemi digitali più complessi. Imparerete come utilizzare un ricevitore/trasmettitore universale asincrono (UART), come interfacciarsi ad un microcomputer basato sul microprocessor 8080, unitamente ai concetti base della programmazione e dell'interfacciamento di un microcomputer. Come lavorare con un UART è trattato nel Bugbook II, mentre le tecniche di interfacciamento dell'8080 sono oggetto dei Bogbooks V e VI.

I Bugbooks V e VI sono orientati ad un insegnamento sperimentale sull'elettronica digitale. Come già detto, il nostro scopo è integrare l'elettronica digitale, l'interfacciamento dei microcomputer, la programmazione dei microcomputer in un singolo corso unificato. I concetti relativi alle tecniche di programmazione ed interfacciamento sono discussi unitamente ai principi di elettronica digitale, verificati sperimentalmente tramite l'utilizzo dei più noti chip, quali il 7400, 7402, 7404, 7442, 7475, 7490, 7493, 74121, 74125, 74126, 74150, 74154, 74181, 74193.

Noi pensiamo che l'elettronica digitale tende sempre più verso l'utilizzo dei microcomputer. Di conseguenza vi sarà un considerevole sforzo in campo didattico per introdurre l'utilizzo dei microcomputer, come del resto sta già accadendo in molte Università ed Istituti Tecnici. Quanto detto va oltre l'ambiente scolastico per interessare professionisti e tecnici desiderosi di aggiornarsi nell'elettronica digitale. I Bugbooks V e VI sono diretti anche a loro.

I Bugbooks V e VI sono da considerare testi per autodidattica. Nei vari capitoli vi sono le risposte a tutte le domande poste e riepiloghi finali per dei concetti trattati. Chi può utilizzare con profitto questi libri? Riteniamo non sia necessario una particolare conoscenza in elettronica. Se siete in grado di organizzare e comprendere i nuovi concetti, di estrapolare "know-how" da nuove situazioni, di realizzare gli esperimenti suggeriti, trarrete il massimo vantaggio dai Bugbooks. I Bugbooks V e VI si configurano molto bene come programma di autodidattica per professionisti che desiderano aggiornarsi nel campo della elettronica digitale.

La nostra esperienza nell'utilizzare i Bugbooks durante i corsi universitari è stata entusiasmante. Stralci dei Bugbooks sono stati tradotti in tedesco, giapponese, francese, italiano, malese. L'attuale traduzione completa in italiano è la prima eseguita in Europa.

Abbiamo poi osservato la carenza di mezzi educazionali in molti campi specifici dell'elettronica digitale; a questo proposito abbiamo definito dei nuovi volumi della serie, quali il "The 555 Timer Applications and Sourcebook, with Experiments" scritto da Howard Berlin, nella versione italiana sempre dalla Jackson Editrice. Altri volumi della serie Bugbooks, sempre pubblicati dalla Jackson in versione italiana, sono: "The Design of Active Filters, with Experiments" e "Designing with Operational Amplifiers, with Experiments".

In collaborazione con il Continuing Education Center and Extention Division del Virginia Polytechnic Institute e State University, vengono tenuti corsi specializzati sull'elettronica digitale e sull'interfacciamento dei microcomputer. In Italia tali corsi sono realizzati in collaborazione con la Jackson Italiana, Elettronica Oggi e la Mipro. I docenti a questi corsi sono Peter R. Rony, David G. Larsen, Paul Field e Frank Settle (Virginia Military Institute; il Dott. Settle è l'editore di Digital Directions, che descrive le tecniche di insegnamento, le applicazioni, i nuovi prodotti con riferimento ai microcomputer ed ai microprocessor). I corsi sono anche tenuti da Jonathan A. Titus e Christofer A. Titus, della Tychon Inc., Blacksburg, Virginia 24060. Jon ha progettato il microcomputer MMD-1 e Chris ha una notevole esperienza nella progettazione hardware e software con sistemi a microprocessor. In Italia i corsi sono tenuti di volta in volta da alcuni dei succitati docenti che si avvalgono della collaborazione dell'Ing. Aldo Cavalcoli, presidente della Mipro, e dal Sig. Valerio Scibilia, responsabile del settore progettazione della Mipro.

Vogliamo ringraziare chi continua ad appoggiare il nostro sforzo educativo. Mr. Murray Gallant e la E-L Instruments che hanno finanziato lo sviluppo del microcomputer MMD-1 di Jon Titus; Mr. Veltri che ha realizzato delle eccellenti fotografie del nostro materiale.

Peter R. Rony, David G. Larsen, Jonathan A. Titus
Blacksburg, Virginia 24060

CAPITOLO 1

CODICI DIGITALI

INTRODUZIONE

Prima di iniziare a programmare il vostro microcomputer, è necessario che conosciate bene come convertire i numeri binari a 8 bit in codice ottale, e *vice versa*, unitamente ad alcuni concetti base sui codici digitali.

OBIETTIVI

Alla fine di questo capitolo sarete in grado di:

- Convertire un numero binario a 8 bit in un numero ottale a tre digit.
- Dare la definizione di codice ottale.
- Dare la definizione di codice binario.
- Convertire un numero ottale a 3 digit in un numero binario.
- Elencare diversi codici digitali.
- Elencare diversi dispositivi bistabili.
- Distinguere tra loro i sistemi di calcolo binari, ottali e decimali.
- Dare la definizione di bit.
- Fare un esempio in cui la quantità, bit al secondo, sia una misura del flusso di informazioni.
- Spiegare cosa significa il termine comunicazione.
- Dare la definizione di codice digitale.

LINGUAGGI, COMUNICAZIONI E INFORMAZIONE

Una delle più importanti caratteristiche proprie di ogni organismo biologico è la possibilità di comunicare con gli altri organismi della stessa specie. Così la comunicazione, che dà a molti organismi un certo vantaggio per la sopravvivenza, nel senso dato da Darwin al termine, è riscontrabile in molte creature multicellulari, partendo dagli insetti fino all'uomo stesso. Al livello degli insetti esistono molti tipi di comunicazione, che vanno dalla danza delle api ad alcune forme di comunicazione per mezzo di agenti chimici. L'uomo può comunicare con l'ausilio dei suoi cinque sensi, come dimostrano gli individui handicappati che hanno perso uno, o più d'uno, dei loro sensi senza per altro essere più sensitivi del normale nei rimanenti.

Se supponiamo che un individuo voglia comunicare con un altro per mezzo del senso dell'udito e mediante l'uso della parola, risulta chiaro che esistono molte sfumature rispetto a come un suono pronunciato venga interpretato dall'individuo che lo ascolta. Nei secoli passati, diversi paesi del globo hanno sviluppato ciascuno un proprio sistema convenzionale per interpretare i suoni e trascriverli. Noi chiamiamo questo sistema convenzionale *linguaggio* o anche *linguaggio straniero.* Esistono migliaia di linguaggi diversi, ma solo un numero relativamente modesto di questi linguaggi può espandersi o ridursi nel corso di molte centinaia di anni. Il latino, una volta la lingua dominante in Europa, è considerata oggi una lingua morta, per quanto sia chiara la profonda influenza esercitata su molte lingue europee.

La comunicazione può essere definita come l'impartire, il tramandare, o lo scambiare idee, conoscenze, informazioni etc. (sia per via orale, scritta o con dei segnali). Questa è una delle attività umane più importanti e tipiche. Come puntualizzato da James Martin nel suo eccellente libro, *"Telecommunications and the Computer*[3]*"*, la capacità delle maggiori *linee di telecomunicazione*, misurata da una quantità detta *bit al secondo*, è avanzata di pari passo con il progresso delle civiltà negli ultimi cento anni. La capacità di queste linee di comunicazione è passata da 1 bit/secondo del 1840 a 50.000.000 di bit/secondo del 1970 p.es., raddoppiando ogni 5,08 anni. Martin ha inoltre focalizzato il fatto che il totale delle conoscenze umane è cambiato molto lentamente prima dell'inizio, relativamente recente, del pensiero scientifico. È stato stimato che dal 1800 il totale delle conoscenze è raddoppiato ogni 50 anni; dal 1950 ogni 10; e dal 1970 si raddoppia ogni 5 anni.

Un linguaggio, che può essere considerato come prodotto della composizione delle parole e delle combinazioni di parole usate da una nazione, un popolo, o gruppo, non è altro che una forma di comunicazione. I geroglifici egiziani, i simboli e le equazioni matematiche, i segnali di fumo degli Indiani americani, il linguaggio dei segni impiegato dai sordi e l'alfabeto Morse sono altre forme di comunicazione usate dall'uomo.

CODICE BINARIO

La "esplosione delle informazioni" avrebbe intasato l'umanità, almeno nei paesi più avanzati, se non si fosse adottato il *sistema di codifica a due stati* per rappresentare tutti i tipi di informazione, cioè i numeri decimali (da 0 a 9), le ventisei lettere dell'alfabeto inglese (dalla A alla Z), operazioni, simboli, regole, e così via. Chiameremo questa codifica col nome di codifica *off-on o codifica binaria.* La codifica binaria può essere rappresentata o espressa da qualsiasi tipo di dispositivo bistabile, come una luce accesa o spenta, un interruttore aperto o chiuso, una scheda meccanografica perforata o non perforata, un nucleo o una parte di nastro o disco magnetici polarizzati "nord" o "sud", due differenti livelli di tensione o corrente,

due differenti frequenze; oppure i simboli astratti 0 (off) e 1 (on). L'importanza del codice binario risiede nel fatto che è possibile costruire dispositivi capaci di cambiare stato molto rapidamente, in tempi minori di 5 ns, o 0,000000005 s. Così un dispositivo può, in linea teorica, trattare, trasmettere o ricevere informazioni alla velocità di 200 milioni di bit al secondo. Trentadue dispositivi di questo tipo che operassero simultaneamente, potrebbero trattare 6, 4 bilioni di bit al secondo. È questa possibilità di base che ha permesso alla società di immagazzinare, trattare e comunicare enormi quantità di informazioni.

BIT

L'unità elementare di informazione è chiamata *bit*, abbreviazione di *binary digit*. Essa è uguale a una decisione binaria, o ad uno dei due possibili ed egualmente probabili valori o stati (come 0 o 1) usati per immagazzinare o trasmette informazioni. Può anche significare "si" o "no"(3). Una informazione è generalmente rappresentata da una serie di bit. Per esempio,

1 0 0 0

è la rappresentazione del numero decimale 8 in codice binario. La serie di bit,

1 1 0 0 0 0 0 1

è la rappresentazione della lettera A nel codice di 8 bit ASCII. Tra breve parleremo di questi due codici.

CODICI DIGITALI

Un *codice digitale* è un sistema di simboli che può rappresentare valori di dati e costruire un linguaggio speciale che un elaboratore o un circuito digitale può interpretare e usare. I codici digitali possono essere considerati come "linguaggi" digitali che permettono di immagazzinare, trattare e comunicare informazioni. Come esistono molte lingue parlate così esistono molti codici digitali diversi. I codici possono essere suddivisi in alcune categorie molto importanti:

Categoria I — Codici impiegati dai circuiti elettronici per effettuare operazioni digitali. Esempio: codice binario.

Categoria II — Codici impiegati per convertire i numeri decimali da 0 a 9 in forma digitale. Esempi: codice binario, codice binario decimale codificato (BCD), codice Gray.

Categoria III — Codici impiegati per convertire numeri decimali, le 26 lettere dell'alfabeto inglese, simboli ed operazioni in forma digitale. Esempi: codice ASCII, codice EBCDIC, codice Baudot.

Categoria IV — Codici di istruzioni impiegati da grandi elaboratori dai mini e dai micro; codici che permettono loro di eseguire una prescritta sequenza di operazioni. Esempi: codice istruzioni per IBM 370, codice istruzioni per PDP 8/E, codice istruzioni per 8080.

All'interno di questa trattazione esamineremo in particolare quattro codici: codice binario, codice binario decimale codificato (BCD), codice ASCII, e codice di istruzioni per il microprocessor 8080.

CODICE BINARIO

Il più semplice codice digitale è quello bistabile, o *binario* codice che consiste di uno stato 0 (off) e di uno stato 1 (on). Chiameremo questi due stati 0 logico ed 1 logico. Nel codice binario, il decimale 0 è rappresentato dallo 0 logico ed il decimale 1 dall'1 logico. Questa affermazione vi risulterà tra breve più chiara. Come rappresentare allora, i numeri decimali più alti, come 3,17, 586 etc. col codice binario? La soluzione sta nell'usare una serie di bit per formare un sistema di calcolo binario che si fonda sulla *base* o *radice* due. Per esempio, il numero binario 11101_2, dove l'indice "$_2$" rappresenta il sistema di conteggio binario, è equivalente a:

$$11101_2 = (1 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = 29_{10}$$

dove dovete ricordare che

2^4 = 16 in notazione decimale = 16_{10}

2^3 = 8 in notazione decimale = 8_{10}

2^2 = 4 in notazione decimale = 4_{10}

2^1 = 2 in notazione decimale = 2_{10}

2^0 = 1 in notazione decimale = 1_{10}

Allora. $11101_2 = 16_{10} + 8_{10} + 4_{10} + 0 + 1_{10} = 29_{10}$

dove l'indice "$_{10}$" associato a questi numeri rappresenta il sistema di calcolo decimale, un sistema basato sulla *base* o *radice* 10. Qui di seguito trovate una breve tavola che vi permetterà di convertire numeri decimali semplici in numeri binari:

numeri decimali	numeri binari
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111
16	10000

Quindi una serie di digit binari, o bit, puo rappresentare uno qualsiasi dei sedici numeri decimali da 0 a 15. I numeri decimali maggiori di 15 richiedono dei bit addizionali,

come mostra la tabella seguente:

numero decimale	numero binario
0	0
1	1
2	10
3	11
4	100
7	111
8	1000 = 4 bit
15	1111 = 4 bit
16	10000
31	11111
32	100000
63	111111
64	1000000
127	1111111
128	10000000 = 8 bit
255	11111111 = 8 bit
256	100000000
511	111111111
512	1000000000 = 10 bit
1023	1111111111 = 10 bit
1024	10000000000
2047	11111111111
2048	100000000000
4095	111111111111
4096	1000000000000
8191	1111111111111
8192	10000000000000
16.383	11111111111111
16.384	100000000000000
32.767	111111111111111
32.768	1000000000000000 = 16 bit
65.535	1111111111111111 = 16 bit

Quindi, un numero binario di otto-*bit*, può codificare 256 numeri decimali diversi, che vanno da 0 a 255_{10}, oppure duecentocinquantasei "entità" diverse, non importa quali esse siano (istruzioni, dispositivi, impulsi, etc.). Dovreste ricordare che l'8080 è un microprocessor che ha indirizzi di memoria a 16-bit, parola d'istruzione a 8-bit e parola dati verso l'I/O a 8-bit. Questo significa che può indirizzare direttamente 65.536 diverse locazioni di memoria, che ha un set di 256 istruzioni diverse e che può generare fino a 256 diversi impulsi di I/O (ingresso/uscita) oppure indirizzare 256 diversi dispositivi esterni.

CODICE OTTALE

Ovviamente è difficile ricordare dei numeri espressi in binario. Per esempio, riuscireste a ricordare il seguente numero binario a 8 bit,

1 0 0 1 1 1 0 1

dopo averlo guardato per un solo secondo?
Svelti, copritelo e guardate altrove! Considerate poi il problema di ricordare una lista di numeri ad 8 bit come la seguente:

1 1 0 1 0 0 1 1

0 0 0 0 0 0 1 0

0 0 1 1 1 1 0 0

1 1 0 0 0 0 1 1

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

Probabilmente avrete concluso che deve esserci un modo migliore per ricordare i numeri binari a 8 bit. Abbiamo usato i numeri a 8 bit perchè li incotrerete di frequente in seguito quando programmerete il microcomputer 8080.

Un sistema per ricordare i numeri binari a più bit è quello di usare il *codice ottale.* Il codice ottale fa riferimento al *sistema di calcolo ottale*, un sistema che si basa sulla *base*, o *radice*, di 8. Il sistema di calcolo ottale consiste di soli 8 simboli differenti: 0, 1, 2, 3, 4, 5, 6 e 7. Come abbiamo già visto per i numeri decimali, è possibile convertire anche i numeri ottali in numeri binari:

numeri ottali	numeri binari
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111
10	001 000
11	001 001
12	001 010
13	001 011
14	001 100
15	001 101
16	001 110
17	001 111
20	010 000
21	010 001
22	010 010
23	010 011
24	010 100
25	010 101
26	010 110
27	010 111
30	011 000
40	100 000
50	101 000
60	110 000
70	111 000
77	111 111

Abbiamo raggruppato i 6 bit dei numeri binari in due gruppi di tre bit per farvi capire più facilmente come si effettua la conversione

da numero-ottale a numero-binario. In realtà lo spazio tra i due gruppi di tre bit non esiste.

Ora torniamo al problema di convertire un numero binario a 8 bit in un codice ottale. Il sistema per eseguire la conversione è costituito da 3 fasi:

1. Scrivere per intero un numero binario a 8 bit.

2. Spezzare questo numero binario a 8 bit in gruppi di tre partendo da destra verso sinistra. Uno dei tre gruppi, quello più a sinistra, è composto di due soli bit. Assumiamo che il terzo bit sia lo 0 logico.

3. Sostituire il *corrispondente digit ottale* 0, 1, 2, 3, 4, 5, 6, 7, ad ogni gruppo di tre bit. Fatto questo avrete convertito un numero binario a 8 bit in un codice ottale a tre digit. Ogni gruppo viene convertito indipendentemente dagli altri.

Per esempio considerate il numero binario a 8 bit

1 0 0 1 1 1 0 1

Primo, spezzate questo numero in tre gruppi

10 011 101

Adesso aggiungete uno 0 logico davanti al gruppo di due bit

010 011 101

Infine, sostituite il digit ottale corrispondente ad ognuno di questi tre gruppi

2 3 5

Questo è il risultato esatto, 235_8, dove l'indice "$_8$" rappresenta il sistema di numerazione ottale. Notate che in un numero binario a 8 bit, il primo digit ottale non può assumere un valore più grande dell'ottale 3. Perchè no?

Altri numeri ottali ed i loro corrispondenti numeri binari a 8 bit sono mostrati qui sotto:

numeri ottali	numeri binari
100	01 000 000
110	01 001 000
111	01 001 001
120	01 010 000
140	01 100 000
170	01 111 000
177	01 111 111
200	10 000 000
240	10 100 000
270	10 111 000
277	10 111 111
300	11 000 000
340	11 100 000
370	11 111 000
377	11 111 111

DOMANDE RIEPILOGATIVE

Le seguenti domande servono ad aiutarvi a ripassare i codici ottali.

1. Cosa è un codice digitale?

2. Elencate vari tipi di codici digitali.

3. Quanti bit ci sono nei seguenti numeri binari?

   a. 11010011

   b. 1000000000000011

   c. 1001

4. A quali numeri decimali corrispondono i seguenti numeri binari?

   a. 11101

   b. 11111111

   c. 1111111111111111

   d. 1001

   e. 11010011

   f. 10011

5. A quali numeri ottali corrispondono i seguenti numeri binari?

   a. 11010011

   b. 00111110

   c. 01110110

   d. 00111100

   e. 11111111

   f. 00110010

   g. 11000011

   h. 00000010

   i. 110

6. A quali numeri binari corrispondono i seguenti numeri ottali?

   a. 323_8

   b. 377_8

   c. 062_8

   d. 076_8

   e. 166_8

   f. 002_8

   g. 5_8

   h. 074_8

   i. 000_8

7. Cosa significano i seguenti indici?

   a. "$_8$"

   b. "$_{10}$"

   c. "$_2$"

8. Date, la definizione dei seguenti termini.

   a. Sistema di numerazione ottale

   b. Bit

   c. Codice binario

   d. Comunicazione

   e. Linguaggio

RISPOSTE

1. Un codice digitale è un insieme di simboli che rappresentano valori di dati e formano uno speciale linguaggio che un computer o un circuito digitale può interpretare e usare.

2. Codice binario. Binario decimale codificato. Codice Gray. Codice ASCII. Codice EBCDIC. Codice di Baudot. Codice istruzioni IBM 370. Codice istruzioni 8080.

3. a. Otto
   b. Sedici
   c. Quattro

4. a. 29_{10}
   b. 255_{10}
   c. 65.535_{10}
   d. 9_{10}
   e. 211_{10}
   f. 19_{10}

5. a. 323_8
   b. 076_8
   c. 166_8
   d. 074_8
   e. 377_8
   f. 062_8
   g. 303_8
   h. 002_8
   i. 6_8

6. a. 11010011_2
   b. 11111111_2
   c. 00110010_2
   d. 00111110_2
   e. 01110110_2
   f. 00000010_2
   g. 101_2
   h. 00111100_2
   i. 00000000_2

7. a. Si riferisce al sistema di numerazione ottale
   b. Si riferisce al sistema di numerazione decimale
   c. Si riferisce al sistema di numerazione binario

8. a. Un sistema di numerazione basato su una base, o radice, di 8
   b. Una unità elementare di informazione. È uguale alla decisione binaria, o alla designazione di uno dei due possibili ed egualmente probabili valori o stati di qualsiasi entità usata per immagazzinare o trasmettere informazioni
   c. Un codice nel quale ogni elemento è uguale a uno dei due diversi stati, comunemente detti 0 ed 1 logico
   d. L'impartire, tramandare o scambiare idee, conoscenze, informazioni etc.. (sia con la parola, lo scritto o per mezzo di segni).
   e. È l'insieme delle parole e dei metodi di combinazione delle parole usate da una nazione, popolo o gruppo

Il vostro modo di rispondere a queste domande può essere sensibilmente diverso. Consultate di nuovo il testo se avete qualche difficoltà a rispondere.

CAPITOLO 2

INTRODUZIONE ALLA PROGRAMMAZIONE DEI MICROCOMPUTER

INTRODUZIONE

Nel seguente capitolo, farete due diversi tipi di esperimenti: (a) esperimenti che richiedono solo la programmazione di microcomputer e (b) esperimenti che richiedono sia programmazione di microcomputer che *interfacciamento,* cioè cablaggio di circuiti che connettono il microcomputer a qualche tipo di dispositivo esterno. Siccome il denominatore comune di tutti gli esperimenti è la programmazione, cominceremo con l'introdurvi ai principi base della programmazione e alle caratteristiche del *linguaggio di programmazione* che userete in questo testo: il set di istruzioni per il microprocessor 8080. Strada facendo definiremo un certo numero di termini importanti, quali *computer, linguaggio macchina, microcomputer* e molti altri. Questa introduzione alla programmazione occuperà un certo numero di capitoli. Infatti abbiamo preferito spiegarvi le nuove istruzioni in gruppi di 5-10 piuttosto che tutte insieme.

OBIETTIVI

Alla fine di questo capitolo sarete in grado di:

- Dare la definizione di computer digitale.
- Dare la definizione di microcomputer.
- Distinguere tra istruzioni per microcomputer scritte in codice binario, codice ottale, o codice mnemonico.
- Distinguere tra rappresentazione mnemonica e linguaggio macchina.
- Dare la definizione di byte.
- Convertire un indirizzo di memoria di 16 bit in un byte di indirizzo HI e LO.
- Convertire istruzioni codificate in binario a 8 bit in istruzioni codificate in ottale e viceversa.
- Distinguere tra memoria di lettura/scrittura e memoria di sola lettura.
- Dare la definizione di memoria.
- Dare la definizione di programma per computer.
- Esprimere le aree delle locazioni di memoria in codice binario o ottale, per il vostro microcomputer di 1 K.
- Identificare i byte a 8 bit in un elenco di numeri binari.

COSA È UN COMPUTER?

Esistono diversi tipi di computer - *computer digitali, computer analogici, computer a fluidi, computer meccanici.* In questi capitoli conoscerete solo *computer digitali*, che comprendono però probabilmente il 99% di tutti i computer oggi in uso. Un computer digitale può essere definito come segue:

Computer digitale Dispositivo elettronico capace di accettare, immagazzinare e manipolare aritmeticamente informazioni e che contenga sia i dati che il programma di controllo. L'informazione è trattata nella forma di digit codificati in binario (0 e 1) e rappresentati da due livelli di tensione.[4]

Ogni dispositivo, di solito elettronico, atto ad accettare informazioni, fare confronti, sommare, sottrarre, moltiplicare, dividere e integrare le informazioni, rappresentate in forma di digit binari (0 e 1) e fornire i risultati di queste operazioni in una forma comprensibile. Le caratteristiche principali di un computer digitale sono quelle di memorizzare, controllare, eseguire operazioni aritmetiche, logiche e di input/output[2].

Bisogna sottolineare che un computer digitale tratta *informazioni binarie,* del medesimo tipo di quelle di cui abbiamo parlato nel capitolo 1. L'informazione binaria è di solito nella forma di *codici digitali:* codici istruzioni; codici impiegati per convertire numeri decimali in forma digitale; codici impiegati dalla circuiteria elettronica per eseguire le diverse operazioni digitali; e infine codici impiegati per convertire l'alfabeto, i numeri decimali, i simboli e le operazioni in forma digitale.

COSA È UN MICROCOMPUTER?

Un *microcomputer* è un computer digitale completo, basato su un *microprocessor.* Un *microprocessor* è un singolo chip di circuito intergrato che possiede almeno il 75% della capacità di un microcomputer digitale nel calcolo e trattamento dati. Di solito non può funzionare senza l'ausilio e l'apporto di altri chip e di memoria. Un circuito integrato è un dispositivo elettronico nel quale gli elementi attivi (per es. i transistor) e gli elementi passivi (per es. i resistori) sono contenuti all'interno di un singolo blocco. Nell'elettronica digitale il termine si applica principalmente a circuiti contenenti elementi semiconduttori[2]. Il microprocessor è il prodotto della tecnologia avanzata dell'industria dei semiconduttori, cioè della capacità che hanno oggi le industrie di costruire miglia di transistori in un solo chip al silicio non più grosso di 0,05 pollici quadrati.

COSA È UN PROGRAMMA PER COMPUTER?

Un *programma per computer* può essere definito come una serie di istruzioni o "statements" preparate in una forma comprensibile al computer; lo scopo di queste istruzioni è quello di conseguire un certo risultato[2]. Questa definizione non implica che il risultato desiderato possa esistere. Per esempio, si può essere interessati solo a sistemare i dati in ingresso in una forma più comoda in modo che possano essere sia immagazzinati che usati come uscita. Coi microcomputer sarete però più interessati a scrivere programmi che controllano le operazioni di una macchina o dispositivo. In una lavatrice domestica potreste pensare di controllare la quantità di acqua da usare, la temperatura dell'acqua nei diversi cicli di lavaggio, il numero e tipo dei cicli da impiegare per ogni tipo di tessuto e la durata di ogni ciclo. Tutte queste cose possono essere fatte con un programma appositamente e correttamente scritto.

ISTRUZIONI

L'istruzione può essere definita come un set di caratteri che definiscono una operazione. Sia singolarmente, che con altre, una istruzione fa sì che un computer digitale esegua una operazione o tratti un determinato dato.

Un *carattere* è un simbolo di un set di simboli elementari come quelli corrispondenti ai tasti di una macchina da scrivere. I simboli comprendono di solito decimali da 0 a 9, le lettere dalla A alla Z, i segni di punteggiatura, il simbolo del dollaro, le virgole, i simboli delle quattro operazioni e ogni altro simbolo che un computer può leggere, memorizzare o scrivere[5]. Nel programmare un computer, non è insolito usare tutta la tastiera della macchina da scrivere, compresi i simboli quali:

@, #, $, %, ¢, &, *, (,), ?, /, e !.

Le istruzioni per un computer si presentano sotto diverse forme. Possono essere *numeri binari,*

11010011
00111110
01110110

numeri ottali

233
076
166

codici mnemonici

OUT 2
MVI A
HLT
MOV B,C
JMP

parole complete,

OUTPUT ACCUMULATOR DATA TO DEVICE # 2

MOVE DATA IN NEXT MEMORY LOCATION TO REGISTER A
HALT

MOVE DATA FROM REGISTER C TO REGISTER B

JUMP UNCONDITIONALLY TO MEMORY LOCATION GIVEN IN THE FOLLOWING TWO INSTRUCTION BYTES

GO DIRECTLY TO JAIL; DO NOT PASS GO; DO NOT COLLECT $ 200

oppure espressioni puramente matematiche

*X = A**2 + B*Y + C*

*X = SQRT (B**2 - 4*A*C)*

I tipi di istruzioni che userete in questi capitoli saranno numeri binari, numeri ottali e rappresentazioni mnemoniche.

RAPPRESENTAZIONI MNEMONICHE

Mnemonico è un termine che descrive qualcosa usato per aiutare la memoria umana[2]. Da questa definizione ricaviamo le seguenti:

Codice mnemonico	Istruzioni scritte in una forma che il programmatore può facilmente ricordare, ma che può essere facilmente convertita in un linguaggio macchina da un computer o da un utente[2].
Linguaggio mnemonico	Un linguaggio di programmazione che è basato su simboli facilmente ricordabili e che può essere assemblato in un linguaggio macchina da un computer[2]
Codice operativo mnemonico, istruzione mnemonica	Istruzioni che sono scritte con notazioni con un senso compiuto, per esempio, ADD, MPY, o STO[2].
Simbolo mnemonico	Un simbolo costruito in modo da aiutare la memoria umana; per esempio, l'abbreviazione MPY usata per "moltiplicare" (inglese "multiply")[2].

In questa trattazione, impiegheremo occasionalmente i codici mnemonici per le istruzioni che userete nel programmare il microcomputer. I codici mnemonici saranno quelli adottati dalla Intel Corporation per il set di istruzioni del suo microprocessor 8080, il quale è composto di 245 diverse istruzioni in linguaggio macchina. Col tempo sarete in grado di effettuare la conversione codice macchina - codice mnemonico e viceversa.

LINGUAGGIO MACCHINA

Un moderno computer digitale elettronico è capace di effettuare manipolazioni usando segnali elettronici binari, di solito due livelli di tensione (+5 volt e potenziale di massa) che rappresentano rispettivamente gli stati logici 1 e 0. Perciò, ogni istruzione è scritta come una serie di uni e zeri che caratterizza una data istruzione e nessun'altra. Diremo la rappresentazione binaria di una istruzione *linguaggio macchina* o *codice macchina.* Per esempio, l'istruzione in linguaggio macchina 00000111_2 ruota di un bit a sinistra il contenuto di un accumulatore presente nel microprocessor 8080. L'istruzione 00001111_2 ruota di un bit a destra il contenuto dell'accumulatore.

Dopo questa serie di capitoli sarete in grado di usare le istruzioni in linguaggio macchina del microprocessor 8080. Queste istruzioni vi saranno proposte in *codice ottale*, in modo che possiate ricordarle più facilmente. Alcuni codici ottali di istruzioni che userete più frequentemente negli esperimenti di programmazione più semplici, comprendono:

323_8	Istruzione di uscita
303_8	Istruzione di salto incondizionato

166_8	Istruzione di Halt (Alt)
074_8	Incrementare di 1 il contenuto dell'accumulatore

In questo paragrafo abbiamo usato alcuni concetti nuovi: *salto incondizionato, uscita, alt incrementare* e *accumulatore.* Riparleremo di tutti quanti tra breve.

UN PROGRAMMA SEMPLICE

Ecco il primo programma che proverete sul vostro microcomputer:

000_8	Nessuna operazione (No operation)
076_8	Muovi il contenuto della prossima locazione di memoria nell'accumulatore
377_8	Byte rappresentante un dato
166_8	Alt

Il programma contiene tre operazioni e un dato. In questo caso il programma è stato scritto in codice ottale, che avete studiato nel capitolo 1. Ma lo stesso programma poteva esser scritto usando il codice binario, come mostrato qui sotto:

00000000_8	Nessuna operazione
00111110_2	Muovi il contenuto della prossima locazione di memoria nell'accumulatore
11111111_2	Byte rappresentante un dato
01110110_2	Alt

Alternativamente il programma poteva esser scritto in codice mnemonico e convertito più tardi in codice macchina con l'aiuto di un programma specifico chiamato *assemblatore.* In questo caso avremmo avuto il seguente programma:

NOP	Nessuna operazione
MVI DATA	Muovi in modo immediato il contenuto della locazione di memoria DATA, nell'accumulatore.
HLT	Alt

Notate che il programma in codice mnemonico è composto da parole o da abbreviazioni di parole (in inglese), come NOP, HLT e MVI.

Come il microcomputer esegue il suo programma? Lo esegue passo passo e la prima istruzione, NOP, sarà la prima ad essere eseguita. Si avrà quindi una sequenza di operazioni del tipo:

1. Il microcomputer esegue l'istruzione NOP. Sta quindi fermo per un *ciclo di istruzione* e poi passa sulla prossima istruzione. L'istruzione NOP ha un uso molto importante, come vedremo in seguito in un successivo programma.

2. All'esecuzione dell'istruzione MVI. che ha il codice ottale 076_8, viene cercato il byte dei dati che si trova nella locazione di memoria immediatamente successiva.
3. Il microcomputer va alla prossima locazione di memoria, dove trova 377. Prende questo valore e lo mette nell'accumulatore del microcomputer.
4. Il microcomputer esegue l'istruzione finale, HLT. Questa istruzione causa l'arresto del computer.

A seconda dei casi, il programma che abbiamo appena visto può sembrare incomprensibile o semplice. Cos'è la memoria? Cos'è un byte? Come si fa a distinguere un byte di una istruzione da un byte di un dato? Cos'è un accumulatore? Sono tutte domande ragionevoli ed alcune ve le sarete poste studiando il programma riportato nelle pagine precedenti. Adesso proseguiamo col rispondere ad alcune di queste domande.

BYTE

Nel microcomputer 8080 un byte è un gruppo di otto bit contigui che occupano una singola locazione di memoria. Per "contigui" intendiamo: adiacenti, vicini, o uno-dopo-l'altro. In realtà un byte può essere un qualsiasi tipo di codice digitale che abbia otto bit. Quindi il numero binario,

0 1 1 0 1 0 0 1

è un byte, mentre il numero binario,

1 0 1 0 0 1

non lo è, infatti contiene solo sei bit adiacenti. Il termine byte è diventato popolare perchè molti computer digitali hanno lunghezza della *parola* multipla di otto bit.

PAROLA

Abbiamo una certa difficoltà a trovare una buona definizione per il termine *parola*. In generale si può dire che la *parola* è il numero di bit che un computer può trattare contemporaneamente. Se poi il numero di bit è otto viene usato il termine byte piuttosto che parola. Il microprocessor 8080 tratta dati di otto bit (byte), ma ha la parola per indirizzamento di memoria di sedici bit. La parola del microcomputer PDP-8 è di dodici bit, e questo significa che il microcomputer durante l'esecuzione dei programmi tratta dodici bit contemporaneamente. Il microcomputer PDP-11 ha la parola di sedici bit e i computer più grossi hanno la parola di trentadue bit. Abbiamo però notato delle inesattezze nell'uso del termine parola, quindi vi consigliamo di stare molto attenti quando incontrate questo termine nei testi specializzati.

MEMORIA

La memoria può essere definita come un dispositivo che è capace di immagazzinare bit logici 1 e 0 in termini di zeri ed uni logici in un modo che permetta poi di accedere ad un singolo bit o a un gruppo di essi[6]. Ci sono molti tipi di memorie che possono effettuare quanto detto, ma nel nostro microcomputer ci sono due tipi di memorie:

Memoria a lettura/scrittura — memoria a semiconduttore, nella quale gli stati logici 0 e 1 possono essere scritti (immagazzinati) e letti (ritrovati). Questa è detta anche *memoria ad accesso casuale* (random), termine che però vi raccomandiamo di non usare.

Memoria a sola lettura memoria a semiconduttore dalla quale i dati, in forma digitale, possono essere ripetutamente letti, ma mai essere scritti come nel caso della memoria a lettura/scrittura(6). La sua abbreviazione è ROM.

In realtà, nel nostro microcomputer, la memoria a sola lettura è un tipo un pò speciale di memoria ed è detta *memoria a sola lettura programmabile e cancellabile* (erasable programmable read-only memory) o EPROM. Nel capitolo successivo parleremo anche della memoria EPROM.

Il punto veramente importante è che le memorie sono composte di dispositivi a semiconduttore; sono relativamente poco costose, non hanno parti meccaniche e non occupano molto posto nella piastra del vostro circuito stampato. Sono, in effetti, uno dei ritrovati che hanno permesso il rapido evolversi della tecnologia dei computer.

Quanta memoria avete a disposizione? Il più semplice microcomputer che potete usare ha 256 byte di memoria a lettura/scrittura e 256 byte di memoria a sola lettura. Siccome un byte contiene otto bit, avete a disposizione nel vostro microcomputer un totale di 4096 bit di memoria e ciò è sufficiente, come vedremo nei prossimi capitoli, per esperimenti di programmazione ed interfacciamento.

È possibile aggiungere al vostro microcomputer 256 byte di memoria a lettura/scrittura e 256 byte di memoria a sola lettura. Il microcomputer con la memoria massima ha una capacità totale di 1024 byte. Viene allora detto microcomputer ad *1 K di memoria,* dove "K" rappresenta circa un migliaio di differenti locazioni di memoria. Un microcomputer di *64 K di memoria* conterrà 65.536 byte, quantità enorme per un sistema così piccolo.

Quanta memoria vi necessita? Dipende da quanto è sofisticato il programma che volete far eseguire. Per molte applicazioni, alcuni "K" sono più che sufficienti. Dobbiamo però qui anticipare una considerazione: i microcomputer che voi userete sono orientati all'apprendimento e al controllo cioè verso programmi molto semplici e di interfacciamento; per questo motivo la memoria necessaria non è molto ampia.

INDIRIZZI DI MEMORIA

Un indirizzo di memoria è definito come una locazione di immagazzinamento di una parola di memoria. Notate che abbiamo detto parola e non byte, infatti per alcuni computer, la parola può essere di 32 bit, e allora ogni locazione di memoria conterrà 32 bit. Nel microcomputer 8080 ogni locazione di memoria contiene un byte, cioè, otto bit.

Nel microcomputer che userete ci sono 1024 diverse possibili locazioni se consideriamo la versione con la massima capacità sulla singola scheda. Questa memoria è suddivisa in gruppi di 256 locazioni ognuno e questo fatto può essere descritto nel modo seguente:

- Gruppo di memoria 0, il primo gruppo di 256 locazioni, ciascuna delle quali contiene otto bit. Questa parte è composta di memoria a sola lettura e contiene il programma base, quello che permette al microcomputer di funzionare. *Voi non potete modificare il contenuto di questa parte di memoria.*

- Gruppo di memoria 1, il secondo gruppo di 256 locazioni, ciascuna delle quali contiene otto bit. Questa zona è riservata per una memoria addizionale a sola lettura, o PROM, che contiene qualsiasi programma vogliate metterci.

- Gruppo di memoria 2, il terzo gruppo di 256 locazioni, ciascuna delle quali contiene otto bit. Questa zona è riservata per una memoria addizionale a lettura/scrittura. È assente nei microcomputer più semplici, ma può esservi aggiunta facilmente e con poca spesa.

- Gruppo di memoria 3, il quarto gruppo di 256 locazioni, ciascuna delle quali contiene otto bit. Questa è la memoria a lettura/scrittura che userete ogni volta che programmerete il vostro microcomputer.

INSIEME DELLE LOCAZIONI DI MEMORIA

Il microcomputer 8080A ha una proprietà molto importante: può indirizzare fino a 65.536 diverse locazioni di memoria, ciscuna delle quali contiene otto bit. Questo perchè il chip ha la parola di indirizzamento di 16 bit. Se fate bene i calcoli, infatti, si possono ottenere 2^{16}, cioè 65.536 combinazioni diverse.

Come abbiamo già detto, il microcomputer 8080A ha solo 1024 locazioni di memoria sulla scheda base. Vi chiederete quindi quali siano, sulle prime 65.536 indirizzabili, quelle disponibili. La risposta è: le prime 1024 locazioni.

Detto in un altro modo, l'insieme degli indirizzi di memoria possibili in questo microcomputer è

da 0000000000000000_2 a 0000001111111111_2

Questa però è una notazione limitativa, infatti è molto difficile da ricordare. Ci sono però sistemi più semplici per identificare le locazioni di memoria e l'insieme di quelle disponibili nel microcomputer e li vedremo in seguito.

INDIRIZZI DI MEMORIA HI E LO

È molto difficile ricordare gli indirizzi di memoria a sedici bit, indubbiamente più difficile dei codici istruzione o dei dati che sono solo di otto bit. Per ricordare questi tipi di indirizzi bisogna pensare che *il microprocessor 8080 tratta gli indirizzi di memoria a sedici bit come due indirizzi di otto bit, un byte di otto bit HI e un byte di otto bit LO.* La loro definizione è la seguente:

Byte di indirizzo HI — Sono gli otto bit più significativi (i primi 8 da sinistra) dei sedici della parola di indirizzo del microprocessor 8080. La sua abbreviazione è H o HI.

Byte di indirizzo LO — Sono gli otto bit meno significativi (gli ultimi 8 da sinistra) dei sedici bit della parola di indirizzo del microprocessor 8080. La sua abbreviazione è L o LO.

Quindi l'insieme delle locazioni di memoria indirizzabili nel vostro microcomputer è:

$$\text{da} \quad \begin{matrix} HI = 00000000_2 \\ LO = 00000000_2 \end{matrix} \qquad \text{a} \quad \begin{matrix} HI = 00000011_2 \\ LO = 11111111_2 \end{matrix}$$

Visto che avete imparato a convertire i numeri binari a otto bit in numeri ottali a tre digit, fate la conversione degli indirizzi di memoria HI e LO che abbiamo visto e otterrete il seguente insieme di locazioni di memoria:

$$\text{da} \quad \begin{matrix} HI = 000_8 \\ LO = 000_8 \end{matrix} \qquad \text{a} \quad \begin{matrix} HI = 003_8 \\ LO = 377_8 \end{matrix}$$

Ricordate la seguente regola: per individuare una *locazione di memoria, dovete specificare sia il byte di indirizzo HI che il byte di indirizzo LO, i quali, insieme, formano una parola di indirizzo di sedici bit.*

DOMANDE RIEPILOGATIVE

Le seguenti domande servono ad aiutarvi a ripassare i concetti introduttivi alla programmazione di un microcomputer.

1. Che differenza c'è tra un computer ed un microcomputer? Naturalmente il termine computer fa riferimento ad un computer digitale.

2. Dite, delle seguenti istruzioni, se sono in codice binario, codice ottale o codice mnemonico.

   a. HLT

   b. 11010011_2

   c. 323_8

   d. MVI

   e. INR

   f. 00111100_2

   g. 166_8

3. Scrivete le seguenti istruzioni binarie in codice ottale

   a. 11010011_2

   b. 01110110_2

   c. 00111100_2

   d. 00110010_2

   e. 00000000_2

   f. 11000011_2

   g. 11111111_2

4. Quale delle seguenti espressioni è un byte?

   a. 1001_2

   b. 011_2

   c. 0000001100000011_2

   d. 1110001101_2

   e. 111000_2

   f. 0100110_2

5. Scrivete i seguenti indirizzi di memoria a 16 bit come byte ottali HI e LO.

   a. 0000001111111111_2

   b. 0000000011111111_2

   c. 0000000111111111_2

   d. 0000001011111111_2

   e. 0000000000000000_2

   f. 0000000100000000_2

   g. 0000001000000000_2

   h. 0000001100000000_2

6. Quali, delle seguenti istruzioni, sono in linguaggio macchina?

   a. NOP

   b. HLT

   c. MVI

   d. INR

   e. 323_8

   f. 166_8

   g. 11010011_2

   h. 000_8

   i. 00111100_2

7. Scrivete l'insieme di memoria dei seguenti gruppi di memoria a semiconduttore del microcomputer 8080, esprimendoli in termini di byte di indirizzo HI e LO.

   a. Le prime 256 locazioni di ROM o EPROM del primo K di memoria.

   b. Le seconde 256 locazioni di ROM o PROM del primo K di memoria.

   c. Le terze 256 locazioni, questa volta di memoria a lettura/scrittura, del primo K di memoria.

   d. Le quarte 256 locazioni, ancora di memoria a lettura/scrittura del primo K di memoria.

8. Date la definizione dei seguenti termini:

   a. Byte b. Indirizzo di memoria c. Codice mnemonico

RISPOSTE

1. Un microcomputer è un computer digitale basato su un solo circuito integrato detto microprocessor. Invece un computer è, almeno fino ad oggi, costruito con più circuiti integrati. Ciascuno dei quali è molto meno complicato di un microprocessor.

2. a. Codice mnemonico
   b. Codice binario
   c. Codice ottale
   d. Codice mnemonico
   e. Codice mnemonico
   f. Codice binario
   g. Codice ottale

3. a. 323_8
   b. 166_8
   c. 074_8
   d. 062_8
   e. 000_8
   f. 303_8
   g. 377_8

4. Nessuna delle espressioni elencate contiene otto bit, quindi nessuna di esse è un byte.

5. a. HI = 003_8 e LO = 377_8
   b. HI = 000_8 e LO = 377_8
   c. HI = 001_8 e LO = 377_8
   d. HI = 002_8 e LO = 377_8
   e. HI = 000_8 e LO = 000_8
   f. HI = 001_8 e LO = 000_8
   g. HI = 002_8 e LO = 000_8
   h. HI = 003_8 e LO = 000_8

6. Le istruzioni e, f, g, h, e i sono in linguaggio macchina

7. a. L'insieme è da HI = 000_8 e LO = 000_8 a HI = 000_8 e LO = 377_8
   b. L'insieme è da HI = 001_8 e LO = 000_8 a HI = 001_8 e LO = 377_8
   c. L'insieme è da HI = 002_8 e LO = 000_8 a HI = 002_8 e LO = 377_8
   d. L'insieme è da HI = 003_8 e LO = 000_8 a HI = 003_8 e LO = 377_8

8. a. Un gruppo di otto contigui che occupano una singola locazione di memoria nel microcomputer 8080.
   b. La locazione di immagazzinamento di una parola di memoria.
   c. Istruzione per computer scritta in una forma tale per cui il programmatore possa facilmente ricordare e che, in seguito, il computer stesso possa facilmente convertire in linguaggio macchina.

CAPITOLO 3

ALCUNE ISTRUZIONI DEL MICROCOMPUTER 8080

INTRODUZIONE

In questo capitolo definiremo alcuni importanti termini quali, *operazione, byte di dati, byte di indirizzo, e codice di dispositivo,* e introdurremo alcune semplici istruzioni del microcomputer 8080A, istruzioni che userete nei programmi proposti nel capitolo 5. Il nostro obiettivo è di introdurre gradualmente l'intero set di istruzioni dell'8080A e proporre programmi che vi permettano di esercitarvi su questi tipi di istruzioni.

OBIETTIVI

Alla fine di questo capitolo sarete in grado di:

- Dare la definizione di operazione
- Rappresentare, in modo semplice, istruzioni a un byte, due byte e tre byte
- Spiegare le differenze tra i seguenti tipi di byte di un programma: codice operativo, byte di dati, codice di dispositivo, byte di indirizzo HI e byte di indirizzo LO
- Dare la definizione di accumulatore
- Dare la definizione di registro
- Dare la definizione di incremento
- Spiegare come operano 6 istruzioni del microcomputer 8080A molto comuni: NOP, HLT, INR A, MVI A, OUT e JMP
- Dire cosa si intende per immediato
- Elencare, per il microcomputer 8080A a 750 kHz, i tempi di esecuzione delle seguenti istruzioni: NOP, HLT, INR A, MVI A, OUT e JMP.

COSA È UNA OPERAZIONE?

Nel capitolo precedente abbiamo definito una *istruzione* come un insieme di caratteri che definiscono, da soli o con altre informazioni, una operazione, e che, nella sua completezza, fa sì che il computer esegua una operazione. *Una operazione* è definita come una specifica azione che il computer eseguirà quando un'istruzione gli dirà di farlo (per esempio, divisione, addizione, sottrazione, esecuzione di OR, etc.). Il numero di operazioni differenti che un computer può eseguire e la velocità con cui lo esegue sono un indice della "potenza" del computer stesso.

Le operazioni che il microcomputer 8080A esegue possono essere suddivise nelle seguenti categorie:

Operazioni di trasferimento di informazioni
Operazioni aritmetiche
Operazioni logiche
Operazioni di subroutine
Operazioni di ingresso-uscita (I/O)
Operazioni di incremento-decremento
Operazioni di salto
Altre operazioni miste

Oppure possono essere suddivise in cinque gruppi come suggerisce la Intel Corporation:

Gruppo di trasferimento dati
Gruppo aritmetico
Gruppo logico
Gruppo di collegamento (BRANCH)
Gruppo di stack, I/O e controllo

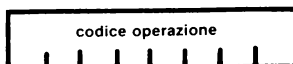
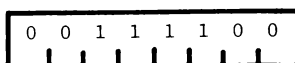
Nella nostra trattazione adotteremo il secondo sistema di suddivisione.

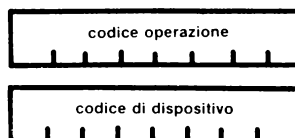
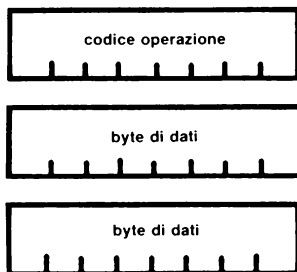
ISTRUZIONI A PIU' BYTE (MULTI-BYTE)

Ricordiamo che un *byte* è un gruppo di otto bit contigui che occupano una singola locazione di memoria. Molte istruzioni, appartenenti al set dell'8080A, richiedono un solo byte, ma alcune hanno bisogno dei due o tre byte successivi per poter essere eseguite. Chiameremo queste ultime, *istruzioni multi-byte.* Le corrispondenti definizioni sono:

Istruzioni a un solo byte	Istruzione consistente di otto bit contigui che occupano una singola locazione di memoria.
Istruzioni a due byte	Istruzione consistente di informazioni che occupano due successive locazioni di memoria.
Istruzioni a tre byte	Istruzione consistente di informazioni che occupano tre successive locazioni di memoria.

Il numero di byte richiesti da una istruzione è strettamente correlata alla complessità dell'istruzione stessa e alle informazioni di cui ha bisogno.

Notate che le istruzioni a due/tre byte usano dei byte che si trovano nelle successive locazioni di memoria. Il primo byte dell'istruzione serve per identificare il tipo stesso di istruzione; una volta che l'istruzione è nota, è automaticamente definito cosa significano i rimanenti byte. Qui sotto sono mostrate alcune semplici rappresentazioni per istruzioni a uno, due, o tre byte. Le istruzioni a un byte richiedono solo otto bit, i quali contengono il *codice operazione*, corrispondente alla specifica operazione del microcomputer 8080A,

Avrete notato che c'è spazio sufficiente per otto bit. Per esempio, l'istruzione di incremento accumulatore, 074_8 o 00111100_2, ha la seguente sequenza di 0 e 1 logici:

Esistono due diversi tipi di istruzioni a due byte, un tipo in cui il secondo byte è un *byte di dati*,

e un altro tipo, in cui il secondo byte è un *byte di codice di identificazione di un dispositivo esterno*

Tra poco spiegheremo cosa si intende per byte di dati e *byte di codice di identificazione di dispositivo*. Esistono inoltre due tipi di istruzioni a tre byte, uno nel quale i due byte finali sono *byte di dati*,

e un altro nel quale il secondo e terzo byte sono byte di indirizzo rispettivamente LO ed HI,

Il concetto di byte di indirizzo di memoria LO e HI è già stato precedentemente discusso.

Se contate gli spazi nelle istruzioni a più byte mostrate qui sopra, noterete che le istruzioni a due byte richiedono 16 bit e che quelle a tre byte ne richiedono 24.

TIPI DI BYTE DI MEMORIA

La memoria di un microcomputer 8080A consiste di una sequenza di locazioni di memoria a otto bit. Tutto quello che il microcomputer fa, e che interessa la memoria, avviene trattando otto bit per volta. I tipi di informazioni che possono essere immagazzinati nella memoria sono:

Codici operativi a otto bit
Byte di dati a otto bit
Codici di identificazione di dispositivo a otto bit
Byte di indirizzo LO a otto bit
Byte di indirizzo HI a otto bit

In un programma per il microcomputer 8080A, vengono immagazzinati contemporaneamente, nella stessa memoria codici istruzione, byte di dati, codici di identificazione e byte di indirizzo; è quindi ragionevole domandarsi come il microcomputer sia capace di distinguerli uno dall'altro.

Si può rispondere che l'ordine con cui appaiono le informazioni, definisce il tipo dell'informazione stessa.

Programmare un calcolatore è un'attività di precisione; un solo errore e il programma non è più in grado di funzionare correttamente. Un programma in un microcomputer inizia a funzionare ad un dato indirizzo di memoria e procede, una operazione dopo l'altra, fino all'indirizzo di memoria finale. I codici operazione nel programma vi dicono cosa aspettarsi; cioè, se il prossimo byte è un byte di dati, byte di indirizzo, byte di codice di identificazione di dispositivo,oppure un altro codice operazione.

CODICE OPERAZIONE

Il primo byte di una istruzione per il microprocessor 8080A è sempre un *codice operazione,* che viene definito come codice a otto bit della specifica azione che il microcomputer 8080A deve eseguire. Questa specifica azione potrebbe essere un trasferimento dati, un'operazione aritmetica, logica, di salto, di stack, di I/O o di controllo. Se desiderate conoscere cosa farà dopo il microcomputer, dovete guardare il codice operazione dell'istruzione seguente. Un sinonimo di codice operazione è *codice istruzione.*

BYTE DI DATI

Un *byte dati* è definito come un numero binario a otto bit che il microprocessor 8080A usa nelle operazioni aritmetiche o logiche, o che immagazzina in memoria. Gli otto bit possono essere in un qualsiasi tipo di codice digitale: codice binario, binario decimale codificato, codice ASCII, etc. Quando usiamo il termine byte di dati, intendiamo dire che gli otto bit non sono un codice operazione, un indirizzo di memoria, o un codice di identificazione di dispositivo. Quando programmerete il microcomputer, vedrete che è molto più conveniente mettere dei dati nel programma, quando e dove volete, piuttosto che far riferimento a locazioni di memoria, magari lontane per gli otto o sedici bit che vi servono.

CODICE DI DISPOSITIVO

Il codice di dispositivo, per un microcomputer 8080A, è il codice di uno specifico dispositivo di ingresso o uscita col quale si vogliono scambiare gli otto bit di informazione e l'impulso di selezione dispositivo. In seguito entreremo più nel dettaglio. È importante notare che si tratta di un codice ad otto bit, il che dà la possibilità di indirizzare 2^8, cioè 256, diversi dispositivi di ingresso e 2^8. cioè 256, diversi dispositivi di uscita. Nel vostro microcomputer, i codici di uscita 000_8, 001_8 e 002_8 corrispondono a tre gruppi di *indicatori* a LED, chiamati memoria HI (HI MEMORY), memoria LO (LO MEMORY) e Dati (DATA).

Avanzando nella lettura di questo testo, vi consigliamo di studiare molto bene il significato di codice di dispositivo e impulso di selezione dispositivo, e il modo di usare quest'ultimo per far funzionare i dispositivi di ingresso-uscita, in sincronia col programma del microcomputer.

BYTE DI INDIRIZZO HI E LO

Vi ricordiamo ancora una volta, che il *byte di indirizzo HI* è composto dagli otto bit più significativi, bit di valore più alto, e che il *byte di indirizzo LO* è composto dagli otto bit meno significativi, bit di valore più basso, dei sedici bit che indirizzano la memoria nel microprocessor 8080A. Siccome l'8080A è un microprocessor a otto bit, che prende dati e istruzioni dalla memoria a blocchi di otto bit per volta, non si può trattare l'indirizzo di memoria di sedici bit se non come una copia di due byte di indirizzo di otto bit.

ALCUNE ISTRUZIONI DELL'8080A

Nel capitolo 5, comincerete a provare dei programmi direttamente sul microcomputer; questi programmi conterranno delle istruzioni a uno, due e tre byte. Tra le varie istruzioni vi saranno anche le seguenti:

000	NOP	Nessuna operazione (No operation)
074	INR A	Incrementa di 1 il contenuto dell'accumulatore
166	HLT	Alt al microcomputer
076	MVI A	Muovi in modo immediato il succcessivo byte di dati nell'accumulatore
323	OUT "n"	Genera un impulso di selezione dispositivo e manda in uscita otto bit dall'accumulatore verso il dispositivo "n"
062	STA	Immagazzina (Store) il contenuto dell'accumulatore nella locazione di memoria indirizzata dai sedici bit dei prossimi due byte di questa istruzione di tre byte
303	JMP	Salta (Jump) incondizionatamente all'indirizzo di memoria dato dai successivi due byte di questa istruzione di tre byte.

Per favore fate attenzione all'elenco precedente, perchè contiene solo codici operazione, cioè solo il primo byte di ogni istruzione. *Nel codice ottale di ogni singola operazione, abbiamo omesso l'indice "$_8$". Da questo punto in avanti, i codici operazione di tre digit, i codici di identificazione di dispositivo, i byte di dati, e i byte di indirizzo di memoria, saranno sempre espressi in codice ottale.*

NOMENCLATURA DEI BYTE DELLE ISTRUZIONI

Tutti i testi editi dalla Intel Corporation usano le seguenti abbreviazioni o simboli per il primo, secondo e terzo byte delle istruzioni a più byte:

<B1>	Primo byte di una istruzione
<B2>	Secondo byte di una istruzione
<B3>	Terzo byte di una istruzione

Per esempio, le istruzioni JMP e STA, che sono di tre byte, possono essere scritte nel seguente modo:

JMP <B2> <B3>

STA <B2> <B3>

Allo stesso modo, le istruzioni MVI e OUT, che sono di due byte, possono essere scritte come

MVI <B2>

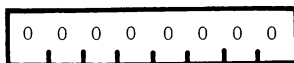
OUT <B2>

ACCUMULATORE

L'accumulatore è un *registro* di otto bit, all'interno del microprocessor 8080A, nel quale vengono posti i risultati di tutte le operazioni aritmetiche e logiche. Un *registro* è un elemento di memorizzazione temporanea e può contenere un solo byte, cioè otto bit, di memoria. Ponete attenzione a quello che potete fare al contenuto dell'accumulatore: per esempio, potete sommare, sottrarre, confrontare dei dati, incrementare, decrementare di 1 il suo contenuto. Potete scambiare il suo contenuto con una locazione di memoria o con dei dispositivi di ingresso-uscita. Potete ruotarne i bit sia verso destra che verso sinistra. Potete eseguire sul suo contenuto delle operazioni logiche , AND, OR e OR-esclusivo. Può darsi che a questo punto non abbiate capito alcune delle cose appena dette, ma abbiate pazienza ancora un po': ne riparleremo tra poco.

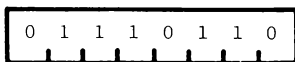
All'interno del microprocessor 8080A, a fianco del registro accumulatore, esistono altri registri; anche di questi parleremo in seguito.

NESSUNA OPERAZIONE: NOP

L'istruzione più semplice del microprocessor 8080A è la NOP, nessuna operazione, ed ha codice istruzione 000,

Non esegue alcuna operazione. Questa istruzione può essere usata ogni volta che volete avere nel programma dello spazio disponibile per poter successivamente introdurre dei byte di istruzione. Nel capitolo 4 avete imparato che il microcomputer lavora a 750 kHz o 750.000 cicli al secondo. *Ogni istruzione del microcomputer richiede del tempo per poter essere eseguita.* Anche se non viene eseguita nessuna operazione, l'istruzione NOP richiede, per essere eseguita, 4 cicli, cioè un tempo totale di 5,333 microsecondi. Il tempo di esecuzione dipende dalla velocità del microcomputer. Se il microcomputer operasse a 2MHz, cioè 2.000.000 di cicli al secondo, l'istruzione NOP richiederebbe, per essere eseguita, solo 2,00 microsecondi. Il normale microprocessor 8080 lavora a 2 MHz, con alcuni circuiti speciali che lavorano ad una velocità leggermente superiore.

Più avanti, in un altro capitolo, definiremo cosa intendiamo per *ciclo.*

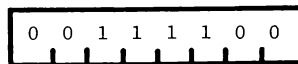
ALT: HLT

Un'altra istruzione molto semplice è quella di Halt, HLT, la quale ha codice di istruzione 166,

Non appena viene eseguita l'istruzione, il microcomputer si ferma. Viene frequentemente usata per permettere al microcomputer di "aspettare un *segnale di interruzione* da un dispositivo esterno. In un computer, un *segnale di interruzione* è una sospensione del normale flusso di una routine, eseguita in modo che il flusso interrotto possa essere più tardi ripreso, dal punto di interruzione. L'istruzione HLT richiede per l'esecuzione, 7 cicli, cioè un tempo di 9,333 μs.

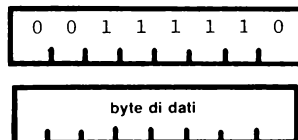
INCREMENTO DELL'ACCUMULATORE: INR A

Il termine *incremento* può essere definito nel seguente modo:

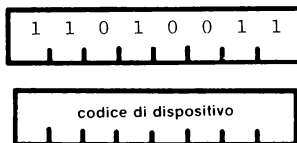
Incremento — Incrementare il valore di una parola binaria. Di solito si intende incremento di 1.

L'istruzione incremento, INR A, che ha come codice istruzione 074,

incrementa di 1 il contenuto del registro accumulatore. L'istruzione INR A richiede per l'esecuzione 5 cicli, cioè 6,667 μs. Quando inizierete a programmare, userete frequentemente questa istruzione, e così potrete "vedere" cosa fa il microcomputer.

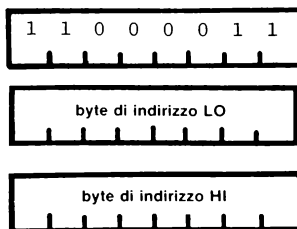
CARICA IN MODO IMMEDIATO NELL'ACCUMULATORE: MVI A

La parola *immediato* si riferisce al fatto che il byte di dati è contenuto all'interno della istruzione stessa, che evidentemente è a più byte. In una istruzione di tipo *immediato,* il byte di dati a otto bit o byte di dati a sedici bit sono acquisiti per mezzo della stessa istruzione a più byte che li contiene come byte <B2> e, eventualmente, byte <B3>. L'istruzione MVI A è di due byte ed ha codice operazione 076,

Il secondo byte dell'istruzione è il byte di dati di otto bit che deve essere caricato nel registro accumulatore. Tutta l'istruzione di due byte richiede per l'esecuzione, 7 cicli, cioè 9,333 microsecondi. Potete constatare che questo è un modo molto conveniente per modificare il contenuto dell'accumulatore; questa istruzione è molto usata.

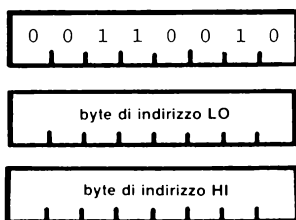
USCITA DATI DALL'ACCUMULATORE VERSO UN DISPOSITIVO PERIFERICO: OUT

Tra tutte le istruzioni esistenti nell'8080A, questa è una delle più importanti. Essa permette di inviare otto bit di dati dall'accumulatore ad un dispositivo periferico che è selezionato per mezzo del secondo byte della istruzione, cioè il *codice di identificazione del dispositivo.* Non bisogna dimenticare che questa istruzione vi permette di generare un singolo impulso di sincronizzazione, detto *impulso di selezione dispositivo,* che sincronizza lo scambio dati tra l'accumulatore e il dispositivo periferico. A suo tempo vedrete come ciò avviene; è una cosa molto ingegnosa. Il codice mnemonico dell'istruzione di uscita è OUT e il codice operazione è 323,

L'istruzione di uscita richiede, per l'esecuzione, 10 cicli, cioè, per un microcomputer a 750 kHz, 13,333 μs. Userete spesso questa istruzione per inviare all'esterno ciò che è accaduto all'interno del microcomputer 8080A. L'istruzione di uscita è un potente strumento per imparare a programmare il microcomputer.

SALTO INCONDIZIONATO: JMP

L'istruzione di salto incondizionato, JMP <B2> <B3>, è una istruzione di tre byte che ha codice operazione 303. Il secondo byte contiene la parte LO dell'indirizzo di memoria e il terzo byte contiene la parte HI,

Quando l'istruzione viene eseguita, il programma salta all'indirizzo di memoria di sedici bit formato dai byte di indirizzo LO e HI. Chiamiamo una istruzione di questo tipo, *istruzione di salto.* Essa permette di arrestare la normale esecuzione sequenziale del programma e saltare in qualsiasi parte della memoria e riprendere, da quel punto, l'elaborazione. Le istruzioni di salto sono molto potenti; permettono infatti di scrivere dei *loop* di programma, cioè dei gruppi di istruzioni che possono essere eseguite ripetute volte. In questo modo si può ridurre di molto la complessità del programma. L'istruzione di salto incondizionato richiede, per l'esecuzione, 10 cicli, cioè 13,333 μs. In questo tempo, vengono eseguiti tutti e tre i byte della istruzione.

MEMORIZZAZIONE DIRETTA DEL CONTENUTO DELL'ACCUMULATORE: STA

L'istruzione di memorizzazione (store) del contenuto dell'accumulatore, STA <B2> <B3>, è una istruzione di tre byte che ha codice operazione 062. Essa vi permette di memorizzare il contenuto dell'accumulatore direttamente in una locazione di memoria, M, determinata dai byte di indirizzo contenuti nel secondo e terzo byte della istruzione. Il secondo byte contiene la parte LO dell'indirizzo e il terzo contiene la parte HI,

Dopo l'esecuzione dell'istruzione il contenuto dell'accumulatore non cambia; esso viene solo ricopiato dall'accumulatore nella locazione di memoria indicata. Questa istruzione richiede, per l'esecuzione, 13 cicli, cioè, per un microcomputer da 750 kHz, 17,333 μs.

DOMANDE RIEPILOGATIVE

Le seguenti domande servono ad aiutarvi a ripassare i concetti discussi in questo capitolo.

1. Qual'è la differenza tra un codice operazione, un byte di dati, un codice di identificazione di un dispositivo, un byte di indirizzo?

2. Dite il codice mnemonico dei seguenti codici operazione espressi in bit:

   a. 01110110

   b. 11010011

   c. 00111110

   d. 11000011

   e. 00110010

   f. 00111100

   g. 00000000

3. Scrivete il codice operazione, nella forma di tre digit ottali, delle seguenti istruzioni dell'8080A:

   a. HLT

   b. JMP

   c. OUT

   d. STA

   e. NOP

   f. INR A

   g. MVI A

4. In una istruzione dell'8080A a più byte, il codice operazione può essere il secondo o terzo byte?

5. Quanto tempo è necessario perchè le seguenti istruzioni siano eseguite se il microcomputer funziona a 750 kHz?

   a. JMP

   b. OUT

   c. MVI A

   d. INR A

RISPOSTE

1. Il codice operazione è un codice di otto bit rappresentante una specifica azione che il microprocessor 8080A eseguirà. Un byte di dati è un numero binario di otto bit che il microprocessor 8080A userà per eseguire operazioni logiche o aritmetiche o di store in memoria. Il codice di identificazione di un dispositivo rappresenta uno specifico dispositivo di ingresso-uscita col quale l'accumulatore del microprocessor 8080A scambierà informazioni di otto bit. Un byte di indirizzo è costituito dagli otto bit più significativi o meno significativi dei sedici bit della parola di indirizzo di memoria.

2. a. HLT
   b. OUT
   c. MVI A
   d. JMP
   e. STA
   f. INR A
   g. NOP

3. a. 166
   b. 303
   c. 323
   d. 062
   e. 000
   f. 074
   g. 076

4. No. Il codice operazione di una istruzione a più byte è sempre il primo byte.

5. a. 13,33 μs
   b. 13,33 μs
   c. 9,33 μs
   d. 6,67 μs

CAPITOLO 4

IL MICROCOMPUTER MMD-1

INTRODUZIONE

Nel seguente capitolo, inizieremo degli esperimenti per dimostrare i concetti sulla programmazione di un microcomputer e sul loro interfacciamento. Per fare questi esperimenti, userete il microcomputer *MMD-1*, basato sull'8080, circuiti integrati, una piastra di montaggio e componenti elettronici vari. Questo capitolo tratterà alcuni concetti base e vi metterà in grado di eseguire in modo corretto i vostri esperimenti. È dato per scontato che il microcomputer sia stato correttamente montato e provato in base alle specifiche impartite.

OBIETTIVI

Alla fine di questo capitolo sarete in grado di:

- Elencare le tensioni di alimentazione richieste dal microcomputer MMD-1.
- Dire quali terminali senza saldatura sono elettricamente connessi tra loro sulla piastra di montaggio (Breadboard).
- Dire la funzione di ciascuno dei sedici tasti della tastiera del microcomputer MMD-1.
- Dire la frequenza e il tempo di un singolo ciclo del microcomputer MMD-1.
- Spiegare la differenza tra una memoria a lettura/scrittura e una memoria a sola lettura programmabile.
- Descrivere le caratteristiche dei quattro blocchi di 256 locazioni di memoria del microcomputer MMD-1.
- Identificare le tre porte di uscita - porta 0, porta 1, porta 2 - del microcomputer MMD-1.
- Verificare le connessioni di alimentazione del microcomputer MMD-1.
- Identificare i seguenti gruppi di otto indicatori a LED del microcomputer MMD-1: indirizzo di memoria HI, indirizzo di memoria LO e REGISTRO DATI.
- Dimostrare il significato delle operazioni S, H, L e dei tasti numerici del Keyboard encoder.
- Caricare ed eseguire un semplice programma di nove istruzioni.

IL MICROCOMPUTER BASE

SCOPO

Lo scopo del microcomputer MMD-1 è quello di darvi l'opportunità di mettere in pratica le conoscenze sulla programmazione e l'interfacciamento che avete appreso. Col microcomputer, sarete capaci di fare degli esperimenti che realizzano praticamente i concetti di cui si è parlato.

Voi farete due tipi di esperimenti: (1) esperimenti che richiedono solo della programmazione, e (2) esperimenti che richiedono sia la programmazione che la costruzione di circuiti digitali di interfaccia con l'utilizzo del Breadboard. Dopo aver eseguito in modo corretto questi esperimenti, sia la parte di programmazione che quella di montaggio, potrete provare il comportamento del microcomputer eseguendo il programma memorizzato.

COME VIENE USATO IL MICROCOMPUTER MMD-1

Il microcomputer MMD-1 è un microcomputer completo, basato sull'8080, di uso didattico, comprendente una tastiera, ventiquattro indicatori a LED, una piastra di montaggio che vi permette di costruire i circuiti di interfaccia per i computer. La piastra di montaggio è un modulo di plastica bianca forata, su cui monterete circuiti integrati, fili, condensatori e resistori: facendo le appropriate connessioni, definirete dei circuiti. Sulla piastra di montaggio vi sono solo due tensioni, +5V e potenziale di massa.

Una tastiera, che si trova sulla destra della piastra di montaggio, vi permetterà di caricare i programmi e di selezionare una specifica locazione di memoria. La tastiera vi permetterà anche di verificare la correttezza del programma introdotto, di eseguirlo e di riportare poi il microcomputer alla condizione iniziale.

I ventiquattro piccoli "oggetti rossi" immediatamente al di sopra della piastra di montaggio sono indicatori a LED.

Un indicatore a LED è un dispositivo elettronico digitale che di solito si accende quando lo stato logico è 1 e si spegne quando è 0. Nel microcomputer MMD-1 gli indicatori, quando vengono accesi, sono rossi. Gli indicatori a LED sono suddivisi in tre gruppi di otto. Il gruppo più a sinistra è l'indirizzo di memoria HI, cioè gli otto bit più significativi dei sedici dell'indirizzo completo. Il gruppo di mezzo è l'indirizzo di memoria LO, cioè gli otto bit meno significativi dei sedici dell'indirizzo completo. Il gruppo di destra è il contenuto della locazione di memoria corrispondente ai sedici bit rappresentati dagli altri indicatori a LED. Discuteremo più avanti i concetti di *indirizzo di memoria, indirizzo di memoria HI, indirizzo di memoria LO,* e *dati* di memoria.

Nelle pagine seguenti sono illustrati alcuni disegni e fotografie del microcomputer MMD-1. In base ad un procedimento detto Silk screening, il microcomputer è stato suddiviso nelle seguenti regioni funzionali, dette anche *blocchi*:

- Regolatore di tensione
- Logica di controllo
- Microprocessor 8080A
- Decodifica di memoria

- Clock
- Memoria
- Decodifica di I/O (ingresso/uscita)
- Buffer
- Bus drivers
- Porta 0
- Porta 1
- Porta 2
- Keyboard encoder

Non è però necessario che voi conosciate questi blocchi funzionali quando inizierete ad operare sul microcomputer. Inizialmente è sufficiente conoscere l'uso del solo Keyboard encoder e delle porte 0, 1 e 2. Man mano che acquisirete esperienza nella programmazione del microcomputer, inizierete ad usare la piastra di montaggio ed alcuni pin localizzati nella regione della decodifica di I/O. Nei capitoli seguenti, studierete più dettagliatamente la circuiteria del microcomputer MMD-1

DESCRIZIONE

- **Alimentazione**

Il microcomputer base è costituito da una piastra a circuito stampato che contiene tutti i componenti, i tasti, gli indicatori a LED. Voi dovete fornire al microcomputer l'alimentazione. Se pensate di fare solamente dei programmi, le alimentazioni necessarie sono le seguenti.

− 5 V : 1,5 A

+ 12 V : 0,15 A

− 12 V : 0,15 A

Ognuna delle tensioni fornite non deve avere più dell'1% di oscillazione. I 5 V devono essere stabilizzati tra 4,75 e 5,25 V ed i 12 V tra 11,25 e 12,75 V.

Volendo interfacciare il microcomputer occorre una tensione supplementare di +5 V per soddisfare le necessità di circuiti integrati esterni. Vi consigliamo di utilizzare un alimentatore che abbia i seguenti requisiti, che sia cioè sufficiente sia per il microcomputer che per la circuiteria di interfacciamento:

+ 5 V : 3,0 A

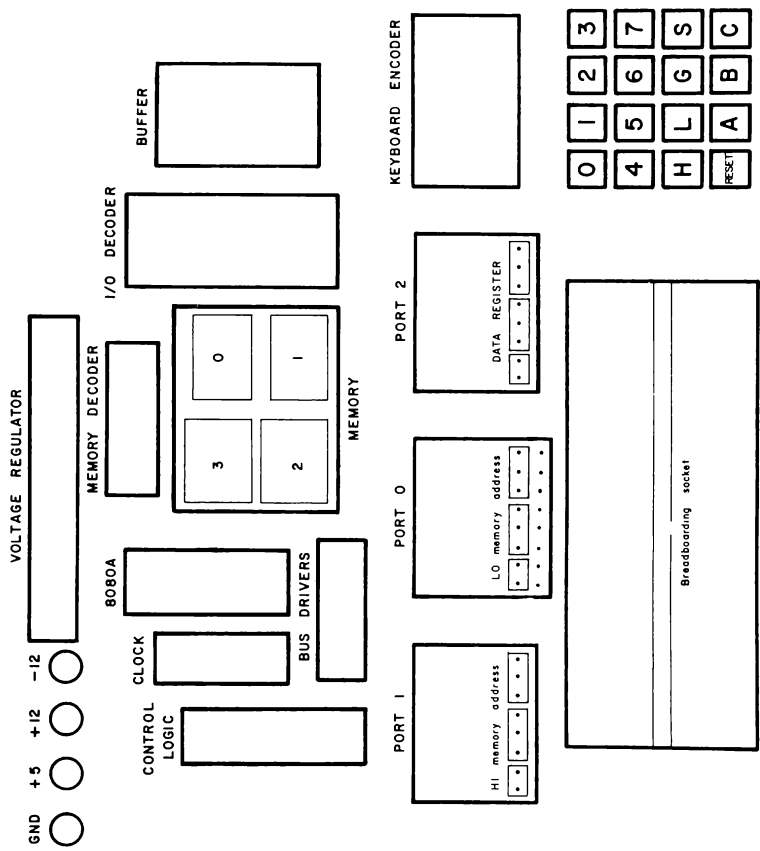



Figura 4-1. Schema a blocchi delle regioni funzionali del microcomputer MMD-1.

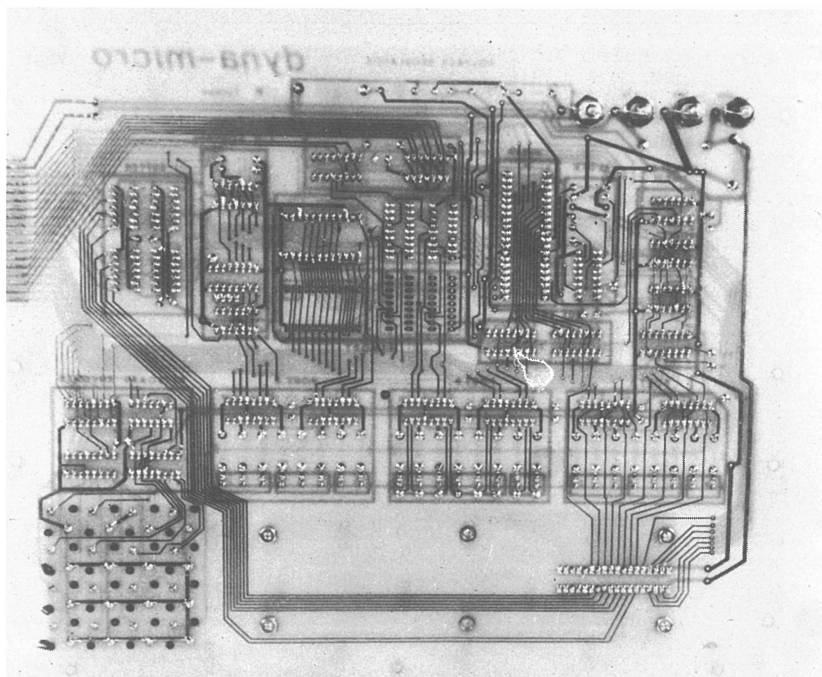
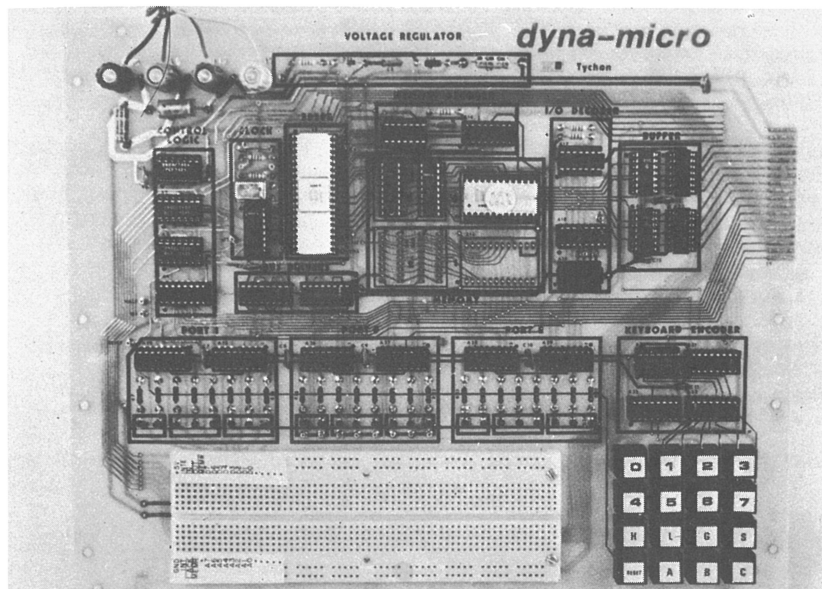



Figura 4-2. Vista superiore e inferiore del microcomputer MMD-1.

● **Piastra (Breadboard)**

La piastra è progettata in modo da poter accogliere i molti esperimenti che farete nei capitoli successivi. I circuiti integrati, i resistori, i condensatori, i fili, e gli speciali dispositivi digitali, detti *funzioni ausiliarie,* saranno collegati direttamente alla piastra.

Le vedute superiore e inferiore della piastra sono illustrate nelle precedenti figure. La piastra contiene 128 gruppi di cinque terminali elettricamente collegati senza saldatura, divisi in due gruppi di 64 ai lati di una banda centrale. Inoltre ci sono 8 gruppi di venticinque terminali sui bordi della piastra. Il terminale "senza saldatura" viene usato poichè è possibile fare connessioni elettriche tra componenti elettronici senza l'ausilio di un saldatore o materiale per saldatura [Nota: Questo è vero dopo l'assemblaggio iniziale se il vostro sistema è stato acquistato in kit.]

Figura 4-3. Vista superiore della piastra.

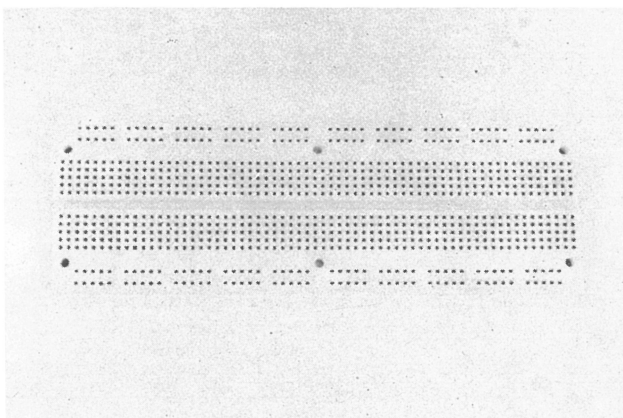
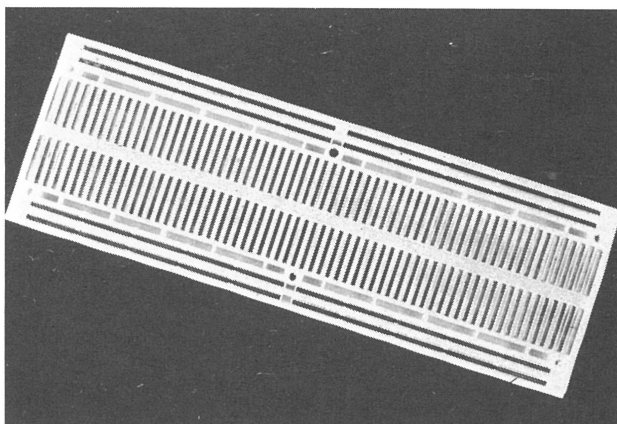



Figura 4-4. Vista inferiore della piastra.

I gruppi centrali di cinque terminali connessi elettricamente sostengono i circuiti integrati e permettono altre quattro connessioni addizionali, da farsi ad ogni pin per i chip più piccoli a 14-pin ed a 16-pin. I gruppi di venticinque terminali connessi elettricamente, ai bordi della piastra, sono collegati o al +5 V e al potenziale di massa. Essi alimentano sia i circuiti integrati che i moduli di funzioni ausiliarie, che descriveremo in un altro capitolo.

● **Keyboard encoder**

Sulla destra della piastra di montaggio vi è una tastiera a 16 tasti. I tasti sono illustrati nella figura 4-5 e ognuno di essi può essere descritto come segue:

Da 0 a 7: Tasti che producono il *digit ottale* indicato ogni volta che sono premuti e rilasciati. Ogni digit ottale è formato di tre bit.

H: La funzione di questo tasto è quella di memorizzare l'informazione presente nel gruppo di indicatori a LED alla destra, dell'indirizzo di memoria HI. Se il microcomputer ha solo il minimo di memoria a lettura/scrittura (256 parole), questo tasto non ha nessun uso.

L: La funzione di questo tasto è quella di memorizzare l'informazione presente nel gruppo di indicatori a LED di destra nei latch dell'indirizzo di memoria LO.

G: Abbreviazione di GO (Vai). La funzione di questo tasto è quella di fare iniziare l'esecuzione di un programma. Prima di premere e rilasciare il tasto G il microcomputer deve essere riportato alla condizione iniziale, altrimenti l'indirizzo di partenza risulta quello indicato dagli indirizzi di memoria HI e LO.

S: Abbreviazione per SEE/STORE (Controlla e memorizza). La funzione di questo tasto è quello di *memorizzare* il dato presente nel gruppo di indicatori a LED di destra (il REGISTRO DATI) nella locazione di memoria HI e LO, detti anche *registri* di indirizzo di memoria HI e LO. Questo tasto può essere usato anche per controllare visivamente se il programma sia stato correttamente introdotto in memoria.

R: Abbreviazione per RESET. La funzione di questo tasto è quella di ripristinare o resettare il microcomputer e gli indirizzi di memoria. Quando viene premuto e rilasciato, gli indirizzi di memoria HI e LO diventano rispettivamente 003_8 e 000_8. Ciò si quò vedere dai gruppi di indicatori a LED a sinistra e al centro.

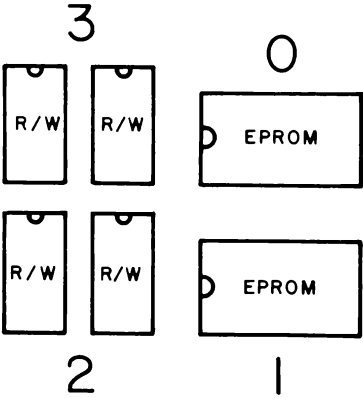
A,B,C: Tasti liberi, che possono essere usati per compiere varie operazioni.
In seguito imparerete come programmare l'uso di questi tasti.

● **Indicatori a LED.**

Ventiquattro diodi ad emissione di luce eseguono la funzione di indicatori. I LED sono posti appena sopra la piastra di montaggio e sono suddivisi in tre gruppi di otto nel modo seguente:

1) Indirizzo di memoria HI; ha una duplice funzione:
a) Mostrare gli otto bit più significativi della parola di indirizzo di memoria a 16 bit;
b) Funziona come Porta 1, accessibile per mezzo dell'istruzione di uscita del microcomputer.

Figura 4-5. Il Keyboard encoder.

Figura 4-6. I blocchi di memoria sulla piastra a circuito stampato.

2) Indirizzo di memoria LO; ha una duplice funzione:

a) Mostra gli otto bit meno significativi della parola di indirizzo di memoria di 16 bit;
b) Funziona come Porta 0 (zero), la quale è accessibile per mezzo dell'istruzione di uscita dal microcomputer.

3) REGISTRO DATI; ha molte funzioni:

a) Mostra il contenuto della locazione di memoria indicata dagli altri sedici indicatori a LED;
b) Funziona come Porta 2, la quale è accessibile per mezzo dell'istruzione di uscita del microcomputer,
c) Funziona come registro di 8 bit temporaneo, il cui contenuto può essere posto nell'indirizzo di memoria LO, nell'indirizzo di memoria HI, o nella locazione di memoria indirizzata dagli altri sedici indicatori a LED. Per quanto riguarda la funzione c), bisogna premere rispettivamente i tasti L, H, o S.

● **Pin della piastra di montaggio.**

I pin sono così situati sulla piastra del microcomputer:

Porta 0:	Otto pin sono situati immediatamente sotto gli otto indicatori a LED corrispondenti all'indirizzo di memoria LO. Questi pin funzionano come uscita della porta 0 e sono accessibili per mezzo dell'istruzione di uscita. I pin permettono di collegare un circuito alla piastra usando la porta 0.
Decodifica di I/O:	Cinque pin sono associati al circuito integrato di decodifica I/O SN74L42. Con l'ausilio di una porta OR o NOR potete generare un impulso selezionatore di dispositivi esterni per una qualsiasi delle seguenti porte di ingresso o uscita: porta 3, porta 4, porta 5, porta 6 o porta 7. I dettagli saranno illustrati in un seguente capitolo.
Altri:	Gli altri pin dislocati sulla piastra adempiono alle seguenti funzioni: HALT, M1, RESET, WAIT, HLDA e O_2 (TTL). Queste ultime saranno trattate in dettaglio man mano che verranno usate.

● **Chip del microprocessor**

Il MMD-1 è un microcomputer basato sull'8080; utilizza l'ultima versione del chip, e cioè l'8080A. All'inizio del 1977 l'8080 era il microprocessor a 8 bit più diffuso sul mercato.

● **Clock**

Il quarzo, che si trova nella sezione di clock della piastra, ha una frequenza di 6,75 MHz. Insieme al quarzo vi è il clock generator and driver 8224 della Intel. Per ottenere la frequenza di clock per l'8080A, la frequenza del cristallo viene divisa per nove. In questo modo, la frequenza di due fasi del clock, O_1 e O_2. è di 750 kHz, cioè 750.000 impulsi di clock al secondo. Ogni singolo *stato* o *ciclo di clock* ha quindi la durata di 1,333 microsecondi. La massima frequenza che può essere applicata ad un 8080A è di 2 MHz. Sfortunatamente, a questa frequenza, i dispositivi di memoria come l'EPROM 1702A, non sono abbastanza veloci. È per questo che è stata scelta la frequenza di 750 kHz.

● Memoria

La memoria del microcomputer è suddivisa in quattro blocchi, ciascuno di 256 locazioni. Come si vede in fig. 4-6 essi sono:

EPROM 0: La Keyboar Executive, o KEX, memoria a sola lettura programmabile, è la memoria che contiene il programma base il quale permette al microcomputer MMD-1 di funzionare. Questa EPROM occupa le prime 256 locazioni di memoria che partono da HI = 000_8 e LO = 000_8.

EPROM 1: Memoria a sola lettura programmabile supplementare da usare per programmi che desiderate non siano cancellati quando togliete l'alimentazione. Questa EPROM occupa le seconde 256 locazioni di memoria che partono da HI = 001_8 e LO = 000_8.

R/W 2: Memoria a lettura/scrittura supplementare; occupa le terze 256 locazioni di memoria che vanno da HI = 002_8 e LO = 000_8.

R/W 3: È la normale memoria a lettura/scrittura usata dal microcomputer. La Keyboard executive PROM fa iniziare l'esecuzione del programma dalla prima locazione di memoria di questo blocco, HI = 003_8 e LO 000_8. Questa memoria a lettura/scrittura occupa le quarte 256 locazioni di memoria nel primo K di sezione di memoria.

Ulteriori particolari riguardo ai blocchi di memoria saranno dati nei successivi capitoli.

● Porte di ingresso-uscita (I/O)

Il microcomputer MMD-1 è provvisto di una porta di uscita a 8 bit a LED, alla quale possono essere collegati dei circuiti, due porte addizionali di uscita a 8 bit con indicatori a LED e cinque segnali decodificati che possono essere usati per generare gli impulsi di selezione dispositivo. Le caratteristiche delle porte di uscita sono le seguenti:

Porta 0: Porta di uscita a 8 bit con 8 pin e otto indicatori a LED.

Porta 1: Porta di uscita con 8 indicatori a LED.

Porta 2: Porta di uscita con 8 indicatori a LED.

Porte da 3 a 7: Porte a 8 bit che possono essere create usando i segnali decodificati presenti nel circuito decodificatore SN74L42, gli impulsi IN e OUT presenti sulla piastra di montaggio ed i segnali da D0 a D7 del data bus bidirezionale a 8 bit presenti anch'essi sulla piastra di montaggio.

Il microcomputer MMD-1 ha anche una porta di ingresso-uscita per la tastiera - la Porta 0 - che userete nei vostri esperimenti.

REGOLE PER ESEGUIRE GLI ESPERIMENTI

Nel seguente capitolo userete il microcomputer MMD-1 per eseguire degli esperimenti che mettono in pratica i principi della programmazione e dell'interfacciamento, ma prima di iniziare, vogliamo raccomandarvi di osservare le seguenti regole base:

1. Pianificate in anticipo gli esperimenti in modo da sapere quali risultati dovreste ottenere.

2. Togliete dalla piastra di montaggio tutti i collegamenti e i componenti rimasti da precedenti esperimenti e quindi non più necessari.

3. IMPORTANTE: Prima di fare un qualsiasi collegamento alla piastra, togliete il collegamento a +5 V dalla piastra di montaggio. Notate che non vi abbiamo detto di togliere l'alimentazione a tutto il microcomputer, perchè, così facendo, si cancellerebbe tutta la memoria a lettura/scrittura.

4. Dopo aver staccato il collegamento +5 V dalla piastra, collegate con attenzione i circuiti di interfaccia al microcomputer. Collegate l'alimentazione a ogni singolo circuito integrato prima di fare qualsiasi altro collegamento.

5. Fate molta attenzione a dove mettete i diversi chip e le funzioni ausiliarie sulla piastra di montaggio; una dislocazione curata di questi dispositivi può ridurre di molto la "giungla" dei fili di collegamento, magari per dei circuiti di modesta complessità.

6. Provate i circuiti che avete collegato per essere sicuri che funzionino bene. *Ponete molta attenzione nell'eseguire i collegamenti di alimentazione ai circuiti integrati.* Infatti, se sono sbagliati, potreste bruciare i chip e cancellare la memoria a lettura/scrittura. Per vedere se i circuiti si scaldano, metteteci sopra un dito; se sono caldi vuol dire che qualcosa è sbagliato.

7. Ricollegate la tensione +5 V quando tutto è già stato provato; a questo punto potete fare il test del dito per verificare se qualche chip si scalda troppo.

8. Una volta terminato l'esperimento, non scollegate i circuiti, ma prima guardate se l'esperimento successivo non richiede l'impiego degli stessi circuiti già collegati.

9. Staccate l'alimentazione principale del microcomputer solo alla fine della giornata di lavoro.

COME SONO PRESENTATE LE ISTRUZIONI PER GLI ESPERIMENTI

Le istruzioni per ogni esperimento sono presentate nel modo seguente:

SCOPO

Tutto quello che viene presentato sotto questo titolo serve a definire lo scopo dell'esperimento, sarebbe quindi meglio, ogni volta che fate un esperimento, ricordare per quale motivo è stato proposto.

CONFIGURAZIONI DEI PIN DEI CIRCUITI INTEGRATI

Sotto questo titolo vengono date le configurazioni dei pin per tutti i circuiti integrati usati negli esperimenti; ci è stato possibile mostrarvele grazie alla gentile concessione della TEXAS INSTRUMENTS INCORPORATED e della INTEL CORPORATION. Nel caso in cui la configurazione sia identica a quella dell'esperimento precedente, può mancare.

SCHEMA DEL CIRCUITO

Ogni volta, vi verrà fornito lo schema di tutto il circuito che userete nell'esperimento e voi dovrete analizzarlo per capirne bene il funzionamento prima di iniziare. Controllate i numeri dei pin di tutti i collegamenti ai circuiti integrati e *ricordate che i collegamenti di alimentazione alle porte (GATES) sono omessi.* Fate molta attenzione alle connessioni con l'alimentazione +5 V e massa perché, se ne dimenticate qualcuna, il circuito corrispondente non funziona.

PROGRAMMA

Vi sarà fornito il programma del microcomputer che voi dovrete caricare all'indirizzo di memoria che vi sarà indicato; se non sarà indicato, allora il programma dovrà essere caricato all'indirizzo di memoria LO per la prima istruzione mentre l'indirizzo HI sarà sempre 003_8. Per trarre vantaggio dal tasto RESET, vi conviene mettere una istruzione JMP all'indirizzo HI = 003_8 e LO = 000_8.

PASSO

Sotto questo titolo, uno per ogni passo sequenziale, cioè PASSO 1, PASSO 2, PASSO 3, etc., sono spiegate in dettaglio le istruzioni su quello che dovete fare in quella particolare parte di esperimento. Durante l'esecuzione dell'esperimento vi saranno poste delle domande cui voi dovrete rispondere subito, apponendo le risposte negli spazi bianchi subito sotto. Dopo aver scritto la risposta, confrontatela con la risposta che vi viene data. Se la vostra risposta non è esatta, vi sarà certamente possibile capirne il motivo; quindi, prima di continuare, correggetela.

DOMANDE

Spesso vi saranno fatte delle domande per controllare (a) quanto avete capito dell'esperimento appena concluso (b) la vostra capacità di anticipare i futuri esperimenti o problemi (c) la vostra abilità nel mettere in relazione il testo con le nozioni apprese direttamente dagli esperimenti o (d) la vostra capacità di fare previsioni o di dare spiegazioni razionali alle più disparate domande che saranno sorte spontanee nello scoprire quanto non vi è stato precedentemente spiegato. Il numero delle domande che vi saranno fatte dipenderà soprattutto dal tipo di esperimento. In alcuni capitoli, le domande saranno raggruppate in un RIEPILOGO, ciascuno dei quali avrà le sue RISPOSTE.

UN ATTIMO DI ATTENZIONE

A tutti i programmatori novizi che ci stanno seguendo vogliamo chiarire molto bene quanto segue:

È impossibile danneggiare un microcomputer con un programma sbagliato.

Potete, per sbaglio, cancellare il contenuto della memoria a lettura/scrittura, ma non potete danneggiare o distruggere il microcomputer avendo degli errori nel programma e tentando di farlo eseguire, quindi rilassatevi e divertitevi col vostro microcomputer, fate pure errori di programmazione e imparate da essi.

Potete però danneggiare il microcomputer MMD-1 se:

- Applicate l'alimentazione in modo errato.
- Toccate, anche incidentalmente, con degli oggetti metallici i collegamenti sulla piastra di circuito stampato. La piastra del microcomputer è provvista di un rivestimento che riduce il problema.
- Sbagliate a collegare i fili in un interfacciamento, quindi state molto attenti nella fase di ingresso dati; *ponete in ingresso i dati con l'ausilio di buffer THREE-STATE.*
- Lo fate cadere.
- Lavorate in un ambiente troppo caldo o corrosivo.
- Cercate di ripararlo senza sapere bene come si fa.

Molti strumenti di laboratorio sono racchiusi in involucri di metallo o di plastica molto dura e sono protetti, fino ad un certo punto, da un uso disattento; invece il microcomputer MMD-1 è privo di involucro per potervi permettere di vedere come è costruito e come funziona; quindi è molto vulnerabile. Noi crediamo sia importante che non vi lasciate intimidire dal microcomputer e che non lo mettiate in un involucro non trasparente.

Allora ricordate: se state solamente eseguendo un programma, non potete, in alcun modo, danneggiare il microcomputer; se invece state programmando e interfacciando il microcomputer dovete stare attenti. In alcuni casi, potete danneggiare con un programma un chip dell'8080A se non usate correttamente un circuito di interfaccia.

INTRODUZIONE AGLI ESPERIMENTI

Questi esperimenti sono proposti per mostrare il funzionamento dei diversi tasti del Keyboard encoder che si trova nell'angolo in basso a destra del microcomputer MMD-1. Per condurre gli esperimenti dovete avere:

- Un microcomputer MMD-1, montato e provato.
- Una opportuna sorgente di alimentazione che fornisca al microcomputer +5 V, −12 V e + 12 V.

In questi primi capitoli non sono previsti esperimenti che necessitano di interfacciamento, quindi non avete bisogno di fili e circuiti integrati.

Il momento più critico per il microcomputer è quando applicate per la prima volta i +5 V, i −12 V e i + 12 V. È possibile che scambiate i collegamenti e quindi applichiate ai terminali tensioni non previste; se vi succede potreste danneggiare il microcomputer. Per favore, state attenti.

Gli esperimenti che farete possono essere sintetizzati nel seguente modo:

Esperimento N°.	Commento
1	Insegnarvi a provare dei collegamenti di alimentazione al microcomputer, applicare la tensione e fare un semplice controllo di questa operazione.
2.	Illustrare come opera il tasto S del Keyboard encoder.
3.	Illustrare come operano i tasti numerici, da 0 a 7, del Keyboard encoder.
4.	Illustrare come operano i tasti H e L del Keyboard encoder.
5.	Illustrare il caricamento di una informazione presente nel REGISTRO DATI in una specifica locazione della memoria a lettura/scrittura.
6.	Illustrare il caricamento e l'esecuzione di un programma molto semplice di nove passi. Voi dovrete solo caricarlo in memoria e provarlo.

ESPERIMENTO N. 1

Scopo

Scopo di questo esperimento è verificare i collegamenti di alimentazione al microcomputer MMD-1, effettuare i collegamenti ed eseguire un controllo molto semplice su questa operazione.

Passo 1

Diamo per sontato che il vostro microcomputer sia già stato correttamente montato e provato e che funzioni secondo quanto specificato; eseguiamo quindi un check sull'alimentazione applicata al microcomputer.
Provate il collegamento di massa tra la piastra a circuito stampato e l'alimentazione. Il terminale di massa è situato sulla piastra a circuito stampato in alto a sinistra ed è il terminale a sinistra più vicino all'angolo della piastra, e, se il microcomputer è stato montato senza errori, la presa a banana deve essere nera.

Passo 2

Provate il collegato a +5 V tra la piastra a circuito stampato e l'alimentazione. Il terminale dei +5 V è vicino a quello di massa e la presa a banana dovrebbe essere blu; ma se fosse di un altro colore non importa, basta che vi accertiate che al terminale sia applicata la tensione corretta.

Passo 3

Provate il collegamento a + 12 V tra la piastra a circuito stampato e l'alimentazione. Il terminale dei + 12 V è situato tra quello dei + 5 V e quello dei − 12 V e dovrebbe essere rosso. Questa tensione alimenta sia il microprocessor 8080A che il clock 8224.

Passo 4

Provate il collegamento a −12 V tra la piastra a circuito stampato e l'alimentazione. Il terminale dei −12 V è il più a destra del gruppo delle quattro prese e dovrebbe essere giallo. I −12 V sono usati dal regolatore di tensione per fornire −5 V all'8080A e −9 V alle EPROM 1702A. L'applicazione delle giuste tensioni al microcomputer MMD-1 è ovviamente un esperimento critico, quindi dovete fare molta attenzione e provare i collegamenti più volte prima di dare tensione.

Passo 5

Date tensione e premete immediatamente il tasto R. I tre gruppi di otto indicatori a LED dovrebbero indicare la seguente informazione logica:

Porta 1 (indirizzo di memoria HI) : 003

Porta 0 (indirizzo di memoria LO) : 000

Porta 2 (REGISTRO DATI): (il contenuto iniziale di questa locazione di memoria).

Se i gruppi di indicatori a LED non visualizzano la configurazione ottale che abbiamo appena indicato, staccate subito le prese di alimentazione del microcomputer; se invece viene visualizzata correttamente passate al prossimo esperimento.

Passo 6

Nel passo precedente vi abbiamo chiesto di applicare tensione al microcomputer e di ripristinare l'indirizzo di memoria al valore HI = 003 e LO = 000 mediante la pressione del tasto R. Premere questo tasto equivale a controllare che il microcomputer funziona correttamente.

Se il vostro microcomputer non funziona in modo corretto eseguite la procedura di controllo totale. Dovete rimuovere, con cautela, l'8080A e il chip 1702A dal loro zoccolino e poi riprovare tutti i livelli di tensione e tutti i circuiti integrati rimasti sulla scheda.

ESPERIMENTO N. 2

Scopo

Lo scopo di questo esperimento è di provare il funzionamento del tasto S del Keyboard encoder.

Passo 1

Date tensione al microcomputer e premete il tasto R. L'indirizzo di memoria diventerà HI = 003 e LO = 000. I LED del REGISTRO DATI mostreranno una configurazione che potrà essere di volta in volta diversa. Quando abbiamo provato noi, è apparso 360 in codice ottale, ed è riapparso ogni volta che abbiamo dato l'alimentazione al microcomputer, quindi non era un dato casuale, ma il vostro microcomputer potrebbe comportarsi diversamente.

Passo 2

Premete il tasto S. Cosa vedete nel registro di indirizzo di memoria LO?

Noi abbiamo osservato che l'indirizzo si è incrementato di 1 diventando 001. Se non è successo anche a voi vuol dire che c'è qualcosa che non funziona e quindi dovete controllare il Keyboard encoder e la parte di ingresso/uscita del vostro microcomputer.

Passo 3

Se il registro LO si è incrementato di 1, premete ancora una volta il tasto S; dovreste notare che si è incrementato di nuovo di 1, diventando 002.

Passo 4

Continuate a premere il tasto S. Ogni volta che lo fate, il registro di indirizzo di memoria LO, aumenterà di 1.

Passo 5

Premete il tasto R e passate all'esperimento seguente.

ESPERIMENTO N. 3

Scopo

Lo scopo di questo esperimento è di provare il funzionamento dei tasti numerici del Keyboard encoder.

Passo 1

Se avete eseguito l'esperimento N. 2, il microcomputer è ormai collegato alla sorgente di alimentazione e l'indirizzo di memoria è riportato sulla posizione HI = 003, LO = 000.

Premete allora tre volte il tasto 0 (zero) e scrivete qui sotto il codice ottale che potete vedere nel REGISTRO DATI (porta 2).

Noi abbiamo visto 000, cioè tutti gli indicatori a LED del REGISTRO DATI spenti.

Passo 2

Premete tre volte il tasto 1 e scrivete il codice ottale che adesso appare sul REGISTRO DATI

Noi abbiamo visto 111

Passo 3

Premete tre volte il tasto 2 e scrivete quello che vedete adesso

Noi abbiamo visto 222

Passo 4

Cosa succederà se premete il tasto 3, poi il 7 e infine lo 0?

Verificate la vostra risposta premendo i suddetti tasti.
Quando abbiamo eseguito noi l'esperimento, sul REGISTRO DATI è apparso 370.

Passo 5

Il proposito di questo esperimento è di dimostrare che i tasti da 0 a 7 sono usati per creare un numero ottale di tre digit, compreso tra 000 e 377, che, per prima cosa, appare sugli indicatori a LED del REGISTRO DATI; successivamente imparerete che questo numero ottale può essere memorizzato, o caricato, nel registro di indirizzo di memoria HI o nel registro di indirizzo di memoria LO, oppure nella locazione di memoria indicata dai due registri di indirizzo.

Per verificare cosa succede quando tentate di introdurre un numero ottale maggiore di 377, premete tre volte il tasto 4 e scrivete quello che è indicato sul REGISTRO DATI.

Noi abbiamo verificato 044; infatti viene perso il bit più significativo dei 9 bit della parola binaria che cercate di creare premendo il tasto 4 tre volte! È impossibile mettere una parola binaria di 9 bit in un registro binario di 8 bit come il REGISTRO DATI.

Passo 6

Controllate, introducendo i numeri ottali tra 400 e 777, che perderete sempre il nono bit, cioè il più significativo.

ESPERIMENTO N. 4

Scopo

Lo scopo di questo esperimento è di provare il funzionamento dei tasti H e L del Keyboard encoder

Passo 1

Date tensione al microcomputer, premete il tasto R; l'indirizzo di memoria sarà riportato al valore HI = 003 e LO = 000

Passo 2

Digitate sulla tastiera il numero 377, cioè premete il tasto 3, poi il 7 e infine ancora il 7. Vedrete che si accenderanno tutti gli indicatori a LED del REGISTRO DATI.

Passo 3

Premete il tasto L. Cosa succede?

Noi abbiamo osservato che l'indirizzo di memoria LO cambia da 000 in LO = 377; infatti era al valore 000 dopo che avevamo ripristinato la condizione iniziale del microcomputer premendo il tasto R.

Adesso l'indirizzo completo di memoria è HI = 003 e LO = 377.

Passo 4

Digitate sulla tastiera il numero 100 e poi premete il tasto L. Cosa succede?

L'indirizzo di memoria LO diventa 100

Passo 5

Digitate sulla tastiera il numero 070 e poi promete il tasto L. Avete visto che l'indirizzo di memoria LO diventa 070?

Passo 6

Digitate sulla tastiera il numero 000 e poi premete il tasto H. Cosa succederà?

Noi abbiamo verificato che l'indirizzo di memoria HI diventa 000. L'indirizzo completo di memoria di 16 bit era HI = 000 e LO = 070, e il contenuto del REGISTRO DATI era 061.

Passo 7

Digitate 002 sulla tastiera e poi premete il tasto H; adesso l'indirizzo di memoria è HI = 002 e LO = 070. Che informazione appare sul REGISTRO DATI?

Noi abbiamo osservato che tutti gli 8 LED sono accesi e questo perché non abbiamo memoria a lettura/scrittura nel blocco 2. *Ogni volta che tentate di indirizzare una locazione di memoria inesistente, otterrete sul REGISTRO DATI il numero 377.*

Passo 8

Digitate il numero 100 sulla tastiera e poi premete il tasto H; adesso l'indirizzo è HI = 100 e LO = 070 che corrisponde ad una locazione di memoria inesistente. Cosa vedete sul REGISTRO DATI?

Noi abbiamo osservato 377 ed è quello che ci aspettavamo.

Passo 9

Digitate 001 sulla tastiera e poi premete il tasto H. Se avete la PROM localizzata nel blocco 1, dovreste vedere sul REGISTRO DATI qualche informazione diversa da 377, se invece la PROM è stata cancellata, dovreste vedere 000. Cosa vedete?

Noi avevamo nel blocco 1 una PROM non programmata, quindi abbiamo osservato sul registro dati il numero 000, corrispondente al contenuto della locazione di memoria HI = 001 e LO = 070.

Passo 10

In questo esperimento avete dimostrato che è possibile caricare un qualsiasi numero ottale a tre digit di otto bit nel registro di indirizzo di memoria LO; infatti per caricare il byte presente nel REGISTRO DATI nell'indirizzo di memoria LO, basta premere il tasto L.

Avete anche dimostrato che è possibile caricare un qualsiasi numero ottale a tre digit di otto bit nel registro di indirizzo di memoria HI; infatti la procedura da seguire è la stessa; per caricare il byte presente nel REGISTRO DATI nell'indirizzo di memoria HI, basta premere il tasto H.

Scegliete alcuni byte di indirizzo di memoria LO e HI e caricateli nei rispettivi registri. Dimostrate che potete caricare qualsiasi indirizzo a otto bit in entrambi i registri.

ESPERIMENTO N. 5

Scopo

Lo scopo di questo esperimento è di mostrare come si possa caricare l'informazione presente nel REGISTRO DATI in una specifica locazione di memoria purché del tipo a lettura/scrittura.

Passo 1

Dopo aver dato tensione al microcomputer, premete il tasto R: l'indirizzo di memoria verrà riportato al valore HI = 003 e LO = 000.

Passo 2

Quale informazione è registrata a questo indirizzo di memoria? Convertite gli otto bit del REGISTRO DATI in un numero ottale a tre digit.

Nel nostro caso, dopo aver dato alimentazione al microcomputer, abbiamo letto nel REGISTRO DATI, per l'indirizzo di memoria HI = 003 e LO = 000, il numero 360.

Passo 3

Premete il tasto 1 tre volte. Osserverete 111_8 nel REGISTRO DATI. Adesso premete il tasto S. Cosa succede?

Il registro di indirizzo di memoria LO si incrementa di 1 e adesso è 001. Quindi nel REGISTRO DATI, appare una nuova informazione.

Passo 4

Gli otto bit che appaiono nel REGISTRO DATI, sono l'informazione memorizzata nell'indirizzo dato dai registri di indirizzo di memoria HI e LO. In questo caso sono HI = 003 e LO = 001; quindi l'informazione che vedete nel REGISTRO DATI *non* è il dato presente nella locazione di memoria HI = 003 e LO =000.

Premete tre volte il tasto 0 e poi il tasto L. Cosa potete osservare adesso?

Noi abbiamo osservato che l'indirizzo di memoria è ridiventato HI = 003 e LO = 000 e che il contenuto di questa locazione di memoria è 111_8. In altre parole, abbiamo caricato 111, in codice ottale, nella locazione di memoria HI = 003 e LO = 000.

Passo 5

Premete tre volte il tasto 3 e poi il tasto S. Il registro di indirizzo di memoria LO si porterà al valore LO = 001.

Passo 6

Premete il tasto R; adesso siete tornati indietro a HI = 003 e LO = 000. Qual'è adesso l'informazione immagazzinata in questa locazione di memoria?

Nel nostro caso è stata 333. In altre parole abbiamo cambiato il contenuto della locazione di memoria HI = 003 e LO = 000 da 111_8 a 333_8.

Passo 7

Caricate 074 nella locazione di memoria HI = 003 e LO = 000 e poi verificate l'esattezza dell'operazione.

Passo 8

Caricate 166 nella locazione di memoria HI = 003 e LO = 000 e dopo controllatelo.

Passo 9

Caricate 323 nella locazione di memoria HI = 003 e LO = 222. Quale procedura dovete ora eseguire per controllare se lo avete fatto correttamente?

Passo 10

Caricate 303 nella locazione di memoria HI = 003 e LO = 350. Controllate ora il dato presente in questa locazione per verificare se è quello corretto.

Passo 11

In questo esperimento avete dimostrato che è possibile caricare un qualsiasi numero di otto bit in una qualsiasi locazione di memoria tra HI = 003 e LO = 000 e HI = 003 e LO = 377. Avete inoltre dimostrato come sia relativamente facile controllare se il caricamento sia avvenuto.

Scegliete alcuni numeri di otto bit e alcuni indirizzi di memoria e dimostrate che potete caricare qualsiasi numero di otto bit in qualsiasi locazione di memoria tra LO = 000 e LO = 377 lasciando costantemente HI = 003.

ESPERIMENTO N. 6

Scopo

Lo scopo di questo esperimento è quello di caricare ed eseguire un semplice programma di nove istruzioni. Inizialmente non è necessario che capiate cosa fa questo programma; basta che lo carichiate e proviate.

Passo 1

Resettate il microcomputer ad HI = 003 e LO = 000.

Passo 2

Caricate 074 nel REGISTRO DATI e premete il tasto S. A questo punto avete "caricato 074 nell'indirizzo di memoria LO = 000" ed inoltre l'indirizzo di memoria si è incrementato di 1 e aspetta di memorizzare la prossima istruzione.

Passo 3

Quale prossima istruzione, caricate 323 nel REGISTRO DATI e premete il tasto S. Avete "memorizzato 323 nell'indirizzo di memoria LO = 001".

Passo 4

Quale terza istruzione, caricate 002 nel REGISTRO DATI e premete il tasto S. Avete "memorizzato 002 nell'indirizzo di memoria LO =002".

Passo 5

Quale quarta istruzione, caricate 315 nel REGISTRO DATI e premete il tasto S. Avete "memorizzato 315 nell'indirizzo di memoria LO = 003".

Passo 6

Memorizzate 277 nell'indirizzo di memoria LO = 004.

Memorizzate 000 nell'indirizzo di memoria LO = 005.

Memorizzate 303 nell'indirizzo di memoria LO = 006.

Memorizzate 000 nell'indirizzo di memoria LO = 007.

Memorizzate 003 nell'indirizzo di memoria LO = 010. Ricordate che nel codice ottale non ci sono nè gli 8 nè i 9 e quindi, dopo l'indirizzo di memoria 007 viene 010.

Passo 7

Premete il tasto R e ripristinate l'indirizzo di memoria LO = 000, mentre l'indirizzo HI dovrebbe essere = 003.

Che informazione è memorizzata in questa locazione di memoria?

Dovrebbe essere il numero ottale 074; se non è così, cambiate il contenuto di questa locazione di memoria in 074.

Passo 8

Premete il tasto R. Adesso il REGISTRO DATI dovrebbe indicare 074. Premete una volta il tasto S, dovreste adesso vedere 323. Premete ancora il tasto S e dovreste adesso osservare 002. Premete ancora una volta il tasto S e adesso dovreste osservare 315 nel REGISTRO DATI.

Passo 9

Nel passo precedente avete controllato l'esattezza del programma che avete caricato. Il programma caricato è il seguente:

Indirizzo di memoria LO	Contenuto del REGISTRO DATI
000	074
001	323
002	002
003	315
004	277
005	000
006	303
007	000
010	003

Accertatevi che tutto il programma sia correttamente caricato nelle posizioni indicate andando a controllare il contenuto di ciascuna di esse. Ogni volta che lo fate, se trovate che il contenuto è corretto, premete il tasto S e continuate con l'indirizzo seguente, se invece non lo è, cambiate il contenuto del REGISTRO DATI col valore giusto e poi premete S.

Passo 10

Premete il tasto R e quindi ritornate ad HI = 003 e LO = 000 preparandovi ad eseguire il programma. Premete il tasto G. Cosa vedete?

Nel nostro caso abbiamo osservato che la porta REGISTRO DATI, porta 2, continua a contare da 000 a 377, dopo di che il ciclo di conteggio si ripete. Gli ultimi due bit alla destra cambiano così rapidamente che non si riesce a vedere altro che un leggero tremolio nella luce dei 2 LED.

Se non vedete la stessa cosa ripetete tutto l'esperimento dal passo 1 finché non memorizzate correttamente il programma e non riuscite ad osservare il risultato voluto.

DOMANDE RIEPILOGATIVE

Le seguenti domande servono ad aiutarvi a ripassare le caratteristiche del microcomputer MMD-1.

1. A cosa serve la piastra sul microcomputer MMD-1?

2. Qual'è la funzione dei seguenti tasti del Keyboard encoder?

   a. H e L

   b. R

   c. G

   d. I tasti numerici da 0 a 7

   e. S

   f. A, B e C

3. Qual'è la differenza tra la memoria a lettura/scrittura e la memoria programmabile a sola lettura?

4. Sul microcomputer MMD-1 esistono tre gruppi di indicatori di monitor a LED. Di questi tre gruppi, qual'è la porta 0, la porta 1 e la porta 2?

   a. Indirizzo di memoria HI

   b. Indirizzo di memoria LO

   c. REGISTRO DATI

5. Che tipo di memoria corrisponde ai seguenti blocchi di 256 locazioni di memoria del micromputer MMD-1?

   a. Blocco 0

   b. Blocco 1

   c. Blocco 2

   d. Blocco 3

6. Cosa succede se per sbaglio, collegate −12 V al terminale +5 V e +12 V al terminale di massa? *Per favore non provate sul microcomputer per vedere cosa succede.*

7. Dove caricate ed eseguite un programma? Elencate le azioni che dovete eseguire.

RISPOSTE

1. La piastra permette di collegare i circuiti di interfaccia ai terminali di ingresso-uscita del microprocessor 8080A. Per effettuare questi collegamenti, avete bisogno di filo, circuiti integrati ed alcune funzioni ausiliarie.

2. a. I tasti H e L caricano un byte dal REGISTRO DATI, rispettivamente nei registri di indirizzo di memoria HI e LO.
   b. Il tasto R resetta il microcomputer, e i registri HI, - LO e DATI. Inoltre restituisce il controllo al programma KEYBOARD EXECUTIVE che si trova nel Blocco 0 di memoria programmabile a sola lettura, cioè l'indirizzo di memoria è riportato ai valori HI = 003 e LO = 000.
   c. Il tasto G fa iniziare l'esecuzione del programma caricato nel microcomputer, dall'indirizzo mostrato dai registri di indirizzo di memoria HI e LO.
   d. Ciascun tasto numerico, quando viene premuto, carica un singolo digit ottale di tre bit nel digit meno significativo della parola di tre digit che si trova nel REGISTRO DATI.
   e. Il tasto S è il tasto SEE/STORE. Vi permette di memorizzare il byte presente nel REGISTRO DATI direttamente nell'indirizzo indicato dai registri di indirizzo di memoria HI e LO; inoltre potete far avanzare un programma passo passo, per provarlo, ricaricando ogni informazione nella sua locazione di memoria.
   f. Tasti di ingressi liberi che possono essere usati per qualsiasi funzione si voglia loro attribuire.

3. Con una memoria a lettura/scrittura si può sia caricare che leggere le informazioni digitali; ma se si toglie l'alimentazione, l'informazione va persa. Con la memoria a sola lettura, è solo possibile leggere informazioni digitali; se togliete l'alimentazione, l'informazione memorizzata non va persa.

4. a. Il registro di indirizzo di memoria HI corrisponde alla porta di uscita N. 1.
   b. Il registro di indirizzo di memoria LO corrisponde alla porta di uscita n. 0
   c. Il REGISTRO DATI corrisponde alla porta di uscita n. 2.

5. a. Memoria programmabile a sola lettura, sia PROM che EPROM. La routine KEYBOARD EXECUTIVE è contenuta in questo blocco di memoria.
   b. La memoria supplementare programmabile a sola lettura, sia PROM che EPROM.
   b. La memoria supplementare a lettura/scrittura.
   d. La memoria a lettura/scrittura che è necessaria per l'esecuzione di un qualsiasi programma del microcomputer. La KEYBOARD EXECUTIVE trasferisce il controllo del programma a questo blocco di memoria.

6. Non lo possiamo sapere finchè non ci proviamo, ma, molto probabilmente, si danneggerebbe il microcomputer perché salterebbero uno o più circuiti integrati. Vi raccomandiamo ancora di non provarci. La cosa importante è stare attenti quando si dà tensione al microcomputer.

7. Passo 1: Ripristinare il microcomputer premendo il tasto R.
   Passo 2: Caricare l'indirizzo di partenza voluto nei registri di indirizzo HI e LO.
   Passo 3: Caricare il programma nelle successive locazioni di memoria partendo da quella indicata nel passo 2.
   Passo 4: Provare la correttezza del programma caricato ritornando all'indirizzo di memoria di partenza e percorrendo il programma passo passo con l'aiuto del tasto S.
   Passo 5: Ritornate all'indirizzo di memoria di partenza e premete il tasto G. A questo punto il programma verrà eseguito.

CAPITOLO 5

ALCUNI SEMPLICI PROGRAMMI PER IL MICROCOMPUTER 8080

INTRODUZIONE

In questo capitolo caricherete ed eseguirete alcuni semplici programmi che usano le istruzioni del microcomputer 8080A di cui abbiamo parlato nel capitolo 3.

OBIETTIVI

Alla fine di questo capitolo sarete in grado di:

- Caricare ed eseguire dei semplici programmi sul microcomputer MMD-1
- Spiegare il modo di funzionamento delle seguenti istruzioni dell'8080A: NOP, INR A, HLT, MVI A, OUT, STA e IMP.
- Mettere in uscita dati di otto bit sia sulla porta 0, porta 1 o porta 2 del microcomputer MMD-1.
- Scrivere un programma che contiene un LOOP (o blocco di istruzioni ripetute più volte) il quale, ad ogni passaggio, incrementa di 1 l'accumulatore.
- Spiegare cosa è un byte di dati.
- Dare la definizione di programma per computer.

COSA È UN PROGRAMMA PER COMPUTER?

Un *programma per computer* può essere definito come una sequenza di istruzioni, le quali, prese nel loro insieme, fanno sì che il computer compia il task voluto. Cos'è un task? Un task è qualsiasi cosa all'interno delle possibilità del computer e dei dispositivi esterni di ingresso - uscita collegati, presupponendo anche che vi sia sufficiente memoria.

Col microcomputer MMD-1, avete a disposizione sulla piastra a circuito stampato base tre blocchi di memoria di 256 byte per memorizzare sia il programma che i dati. Questa memoria è sufficiente per un programma di ragionevole complessità, ma non vi consente di usare il linguaggio BASIC o FORTRAN nè aritmetica a virgola mobile; sarebbero necessari almeno qualche migliaio di byte di memoria.

RIEPILOGO DI ALCUNE ISTRUZIONI DELL' 8080A

Nel capitolo 3 abbiamo parlato delle seguenti struzioni dell'8080A:

000	NOP	Nessuna operazione
074	INR A	Incrementa di 1 il contenuto dell'accumulatore
166	HLT	Alt del microcomputer
076	MVI A	Metti (Move) il byte di dati seguente nell'accumulatore in modo immediato
<B2>		
323	OUT "n"	Genera un impulso di selezione di dispositivo esterno e manda in uscita otto bit dall'accumulatore al dispositivo "n", dove "n" può andare da 000 a 377
<B2>		in codice ottale
062	STA	Memorizza (Store) il contenuto dell'accumulatore nella locazione di memoria di 16 bit indirizzata
<B2>		dai successivi due byte di questa istruzione
<B3>		di tre byte.
303	JMP	Salta (JUP) in modo incondizionato all'indirizzo di memoria dato dai prossimi due
<B2>		byte di questa istruzione di tre byte
<B3>		

Al contrario di quanto illustrato nel capitolo 3, questo elenco contiene sia i codici operazione che tutti i byte addizionali di ogni istruzione.

COME VENGONO PRESENTATI I PROGRAMMI?

In questi capitoli presenteremo programmi nel seguente modo (indicato anche come "LISTING"):

Indirizzo di memoria LO	Byte di istruzione	Codice mnemonico	Commento
011	076	MVI A	Mette il byte di dati seguenti nell'accumulatore
012	111	111	Byte di dati
013	166	HLT	Alt del microcomputer.

Questo è il primo programma che eseguirete. Sul listing del programma vi daremo le seguenti informazioni:

- **Indirizzo di memoria LO**

  Nella colonna di sinistra compare l'indirizzo di memoria di ogni byte di istruzione e si *assume che l'indirizzo HI sia 003.*

- **Byte di istruzione**

  Nella colonna vicina sulla destra compare il contenuto dell'indirizzo di memoria LO corrispondente. Un byte di istruzione può essere un codice operazione, un codice di selezione dispositivo, un byte di indirizzo o un byte di dati. Se il programma è senza errori, il microcomputer è in grado di comprendere di quale tipo sia il byte dell'istruzione che segue.

- **Codice mnemonico**

  Nella colonna sotto questo titolo compare il codice mnemonico di ogni istruzione ed è quello usato dalla Intel Corporation in tutte le sue pubblicazioni. Siccome il microprocessor del tipo 8080 avrà probabilmente una grande diffusione e siccome il suo set di istruzioni sta diventando standard nell'industria elettronica, vi conviene cercare di imparare a memoria questi codici; ciò vi tornerà comodo quando scriverete altri programmi.

- **Commenti**

  In questa colonna spiegheremo brevemente cosa fa ogni singola istruzione. Man mano che proseguirete in questi capitoli vi accorgerete che i commenti si faranno sempre più brevi.

SCELTA DELL'INDIRIZZO DI PARTENZA DEL PROGRAMMA

Il valore dell'indirizzo di partenza del vostro programma non è determinante purché sia all'interno della memoria a lettura/scrittura disponibile sul vostro microcomputer MMD-1. L'indirizzo di partenza più usato è HI = 003 e LO = 000 per il fatto che si può andare a questo indirizzo premendo solamente il tasto R. Quando dovete provare un nuovo programma, se volete usare ancora l'indirizzo di partenza HI = 003 e LO = 000, basta che lo carichiate esattamente sopra il precedente.

Dato che questa serie di capitoli ha lo scopo di insegnare la programmazione e l'interfacciamento, abbiamo deciso di fare in un altro modo, cioè invece di far partire tutti i programmi dallo stesso indirizzo di memoria, li carichiamo sequenzialmente uno dopo l'altro e quindi non dovrete cancellare nessuno dei programmi scritti nella esercitazione di laboratorio in corso. E per di più farete esperienza nello scrivere programmi che hanno diversa dislocazione all'interno della memoria.

Un altro vantaggio collegato al posizionamento sequenziale dei programmi è la *possibilità di metterli tutti nella memoria PROM.* Una parte della vostra esperienza di studio sul microcomputer è basata sul caricare e provare singoli programmi, ma, dopo un pò di tempo, scoprirete che è molto più divertente provarne di nuovi, piuttosto che caricarli e controllarli.

PRIMO PROGRAMMA

Il primo programma che proverete è il seguente:

Indirizzo di memoria LO	Byte di istruzione	Codice mnemonico	Descrizione
011	076	MVI A	Metti in modo immediato il byte dati seguente nell'accumulatore.
012	111	111	Byte dati
013	166	HLT	Alt del microcomputer

Provando il programma, metterete il byte dati contenente 111 nell'accumulatore, dopo di che il microcomputer si fermerà.

Come farete a stabilire se è successo qualcosa? Non potete farlo con questo programma poichè non è previsto alcuna uscita, e quindi, esteriormente, non accadrà nulla. Fate esperienza di questo difetto di programmazione sull'MMD-1: *Dovete sempre prevedere una qualche forma di uscita in modo da capire se il programma è stato eseguito.*

SECONDO PROGRAMMA

Per dare un'uscita al primo programma basta aggiungere due istruzioni di un byte ciascuna:

Indirizzo di memoria LO	Byte di istruzione	Codice mnemonico	Descrizione
014	076	MVI A	Carica in modo immediato il byte dati seguente nell'accumulatore
015	111	111	Byte dati
016	323	OUT	Poni in uscita il contenuto dell'accumulatore sulla porta 1
017	001	001	Codice di dispositivo della porta 1
020	166	HLT	Alt del microcomputer

Quando eseguite questo programma, il byte di dati 111 va nell'accumulatore. Quest'ultimo pone poi in uscita il suo contenuto sulla porta 1. Dopo aver fatto tutte queste cose il microcomputer si ferma e voi potete vedere che il programma è stato completato poiché appare 111 come uscita della porta 1.

VARIAZIONI AL SECONDO PROGRAMMA

Nel secondo programma è possibile cambiare sia il contenuto del byte di dati che il codice del dispositivo di uscita. Per evidenziare queste possibilità è meglio scrivere il programma nel modo seguente:

Indirizzo di memoria LO	Byte di istruzione	Codice mnemonico	Descrizione
014	076	MVI A	Metti in modo immediato il byte dati seguente nell'accumulatore.
015	<B2>	<B2>	Byte dati
016	323	OUT	Metti in uscita il contenuto dell'accumulatore sul dispositivo specificato dal codice selezione dispositivo che segue
017	<B2>	<B2>	Codice di dispositivo
020	166	HLT	Alt del microcomputer

Il primo byte <B2> può essere un qualsiasi byte di dati tra 000 e 377 mentre il secondo byte <B2> deve essere un codice di selezione dispositivo che esiste ed è accessibile sul vostro microcomputer MMD-1. Il codice di selezione della porta 0 è 000, quello della porta 1 è 001 e quello della porta 2 è 002. Nel capitolo seguente utilizzerete i codici di dispositivo da 003 a 007.
Il programma modificato eseguirà le seguenti operazioni. Il byte di dati <B2> viene messo sull'accumulatore; il contenuto dell'accumulatore viene messo in uscita sul dispositivo che ha il codice specificato dal byte di codice di selezione <B2>; alla fine il microcomputer si ferma. *Per eseguire il programma una seconda volta, basta premere il tasto RESET per ridare il controllo alla Keyboard executive (KEX).*

TERZO PROGRAMMA

Il terzo programma è molto simile al secondo, c'è solo una istruzione in più, un incremento di 1 dell'accumulatore.

Indirizzo di memoria LO	Byte di istruzione	Codice mnemonico	Descrizione
021	076	MVI A	Metti in modo immediato il byte dati successivo nell'accumulatore
022	000	000	Byte dati
023	074	INR A	Incrementa di 1 il contenuto dell'accumulatore
024	323	OUT	Poni in uscita il contenuto dell'accumulatore sulla porta 1
025	001	001	Codice di dispositivo della porta 1
026	166	HLT	Alt

Con questo programma mettete il byte dati 000 nell'accumulatore. Si aggiunge poi 1 a questo valore ottenendo 001 nell'accumulatore. Questo nuovo valore viene messo in uscita sulla porta 1 dell'MMD-1. Infine il microcomputer viene fermato.

QUARTO PROGRAMMA

Nel quarto programma sostituiamo l'istruzione di ALT (HLT) con una istruzione di salto (JMP).

Indirizzo di memoria LO	Byte di istruzione	Codice mnemonico	Descrizione
027	076	MVI A	Metti in modo immediato il byte dati seguente nell'accumulatore
030	000	000	Byte dati
031	074	INR A	Incrementa di 1 il contenuto dell'accumulatore
032	323	OUT	Poni in uscita il contenuto dell'accumulatore sulla porta 1
033	001	001	Codice di dispositivo della porta 1
034	303	JMP	Salta incondizionatamente all'indirizzo di memoria contenuto nei prossimi due byte
035	031	–	Byte di indirizzo LO
036	003	–	Byte di indirizzo HI

Questo programma inizia come il terzo programma. Il byte dati 000 viene posto nell'accumulatore. Il contenuto di quest'ultimo è incrementato di 1 diventando 001. Questo nuovo valore è posto in uscita sulla porta 1. Da questo punto in avanti iniziano le differenze dal programma precedente. Infatti, invece di fermarsi, il microcomputer salta all'indirizzo HI = 003 e LO = 031 dove inizia nuovamente l'esecuzione. Il contenuto dell'accumulatore viene incrementato di 1, da cui il valore 002 che va in uscita sulla porta 1. Poi viene eseguito un altro salto, viene incrementato di 1 il contenuto dell'accumulatore e messo ancora in uscita sulla porta 1. Quello che abbiamo descritto viene detto *loop,* definito come una sequenza di istruzioni che vengono eseguite ripetutamente fino a che non si verifichi una condizione di interruzione. Nel caso in esame, la condizione di interruzione è rappresentata dal segnale che invia il tasto R quando lo premete.

La sequenza di istruzioni del LOOP è formata dai byte di memoria compresi tra LO = 031 e LO = 036. Avrete una certa difficoltà a distinguere gli effetti prodotti dal secondo e dal quarto programma. In entrambi i casi infatti, restano accesi tutti i LED della porta 1. L'unica differenza sta nel fatto che, nel quarto programma, il registro di diodi emettitori cambia molto rapidamente, un digit binario alla volta, tra i valori 000 e 377. Il processo avviene tanto rapidamente che voi percepite a media della luminosità di ogni LED.
Ognuno di essi sembra acceso, ma non con l'intensità che caratterizza solitamente gli indicatori a LED.

QUINTO PROGRAMMA

Nel quinto programma ponete in uscita il contenuto dell'accumulatore su tre differenti porte: la porta 0, la porta 1 e la porta 2. Per il resto il quinto programma è identico al quarto.

Indirizzo di memoria LO	Byte di istruzioni	Codice mnemonico	Descrizione
037	076	MVI A	Metti in modo immediato il byte dati seguente nell'accumulatore
040	252	252	Byte dati
041	000	NOP	Nessuna operazione
042	323	OUT	Poni in uscita il contenuto dell'accumulatore sulla porta 0
043	000	000	Codice di dispositivo della porta 0
044	323	OUT	Poni in uscita il contenuto dell'accumulatore sulla porta 1
045	001	001	Codice di dispositivo della porta 1
046	323	OUT	Poni in uscita il contenuto dell'accumulatore sulla porta 2
047	002	002	Codice di dispositivo della porta 2
050	303	JMP	Salta incondizionatamente all'indirizzo di memoria contenuto nei due byte successivi
051	041	–	Byte di indirizzo LO
052	003	–	Byte di indirizzo HI

Le tre istruzioni di uscita sono situate agli indirizzi di memoria LO = 042, 044 e 046. I codici di dispositivo 000, 001 e 002 sono rispettivamente gli indirizzi LO = 043, 045 e 047. Quando eseguirete questo programma, noterete che i LED delle tre porte rappresentano il numero 252, sono cioè accesi alternativamente, uno si e uno no.

SESTO PROGRAMMA

Nel sesto programma userete una nuova istruzione, la STA, la quale mette il contenuto dell'accumulatore in una locazione di memoria invece che verso una porta di uscita.

Indirizzo di memoria LO	Byte di istruzioni	Codice mnemonico	Descrizione
053	076	MVI A	Metti in modo immediato il contenuto del byte dati seguente nell'accumulatore
054	222	222	Byte dati
055	062	STA	Memorizza (Store) il contenuto dell'accumulatore nell'indirizzo di memoria contenuto nei due byte successivi

056	061	—	Byte di indirizzo LO
057	003	—	Byte di indirizzo HI
060	166	HLT	Alt del microcomputer
061	?		Locazione di memoria in cui memorizzare il contenuto dell'accumulatore.

Il programma prima mette il byte dati 222 nell'accumulatore. Poi mette il contenuto dell'accumulatore nella locazione all'indirizzo di memoria HI = 003 e LO = 061. Infine il microcomputer si ferma. Quando avrete resettano il microcomputer potrete andare a questa locazione di memoria e controllare se il contenuto è effettivamente 222.

INTRODUZIONE AGLI ESPERIMENTI

I seguenti esperimenti vi permettono di eseguire alcuni dei programmi che abbiamo descritto in questo capitolo vi farete esperienza nel caricare ed eseguire programmi.

Gli esperimenti che eseguirete possono essere riassunti nel seguente modo:

Esperimento N.	Commento
1	Mostrate l'esecuzione di un programma che non prevede alcuna uscita nè su una porta nè in memoria.
2	Mostrare l'esecuzione di un programma che contiene una istruzione OUT, la quale permette il trasferimento di informazioni dell'accumulatore a una delle porte di uscita del microcomputer MMD-1.
3	Mostrare l'esecuzione di un programma che contiene un piccolo LOOP.
4	Mostrare l'esecuzione di un programma che carica un byte di dati in una prefissata locazione di memoria. Dopo che il programma è finito, esaminate il contenuto della locazione di memoria nella quale dovrebbe trovarsi il byte dati.

ESPERIMENTO N. 1

Scopo

Lo scopo di questo esperimento è di caricare ed eseguire il "primo programma" descritto in questo capitolo.

Programma

Indirizzo di memoria LO	Byte di istruzione	Codice mnemonico	Descrizione
011	076	MVI A	Metti in modo immediato il byte dati seguente nell'accumulatore
012	111	111	Byte dati
013	166	HLT	Alt del microcomputer

Passo 1

Premete il tasto RESET, R, e resettate il microcomputer a HI = 003 e LO = 000.

Passo 2

Caricate 011 nel REGISTRO DATI e premete il tasto L. Adesso siete all'indirizzo di memoria HI = 003 e LO = 011.

Passo 3

Caricate 076 nel REGISTRO DATI e premete S. Caricate 111 nel REGISTRO DATI e premete ancora S. Infine caricate 166 nel REGISTRO DATI e premete una terza volta S. Fatto ciò avete "caricato in memoria il programma".

Passo 4

Caricate 011 nel REGISTRO DATI e premete il tasto L. A questo punto siete pronti ad eseguire il programma appena caricato che inizia appunto all'indirizzo di memoria HI = 003 e LO = 011. Premete il tasto G.

Ora il programma è finito, cosa vedete con riferimento ai LED?

Noi non abbiamo osservato alcun cambiamento e questo perchè non abbiamo messo nel programma nessuna istruzione di uscita.

Continuate con l'esperimento n. 2 dove aggiungete una istruzione di uscita al programma appena fatto.

ESPERIMENTO N. 2

Scopo

Lo scopo di questo esperimento è caricare ed eseguire il "secondo programma" descritto in questo capitolo. Questo programma contiene una istruzione di uscita, così, dopo che è stato eseguito, potrete vedere cosa ha fatto. Il byte di dati e il codice dispositivo non sono gli stessi del secondo programma.

Programma

Indirizzo di memoria LO	Byte di istruzione	Codice mnemonico	Descrizione
014	076	MVI A	Metti in modo immediato il contenuto del byte dati seguente nell'accumulatore
015	155	155	Byte dati
016	323	OUT	Poni in uscita il contenuto dell'accumulatore sulla porta 0
017	000	000	Codice di dispositivo della porta 0
020	166	HLT	Alt del microcomputer

Passo 1

Premete il tasto RESET. Caricate 014 nel REGISTRO DATI e premete il tasto L

Passo 2

Caricate in memoria il programma partendo dall'indirizzo di memoria HI = 003 e LO = 014.

Passo 3

Caricate 014 nel REGISTRO DATI e premete il tasto L. Premete il tasto G. Cosa vedete sui LED della porta 0, cioè sul registro di indirizzo LO che ora stiamo usando come porta di uscita?

Noi abbiamo osservato il valore 155. Se lo vedete anche voi vuol dire che il vostro programma funziona bene.

Passo 4

Quali modifiche bisogna apportare al programma per avere in uscita sulla porta 2 un byte dati pari a 370?

Bisogna cambiare il byte di istruzione all'indirizzo di memoria LO= 015 in 370 e il byte di istruzione a LO = 017 in 002.

ESPERIMENTO N. 3

Scopo

Lo scopo di questo esperimento è di caricare ed eseguite il "quarto programma" di questo capitolo. Il byte di dati e il codice di dispositivo non sono gli stessi del quarto programma.

Programma

Indirizzo di memoria LO	Byte di istruzione	Codice mnemonico	Descrizione
027	076	MVI A	Metti in modo immediato il contenuto del byte dati seguente nell'accumulatore
030	377	377	Byte dati
031	074	INR A	Incrementa di 1 il contenuto dell'accumulatore
032	323	OUT	Poni in uscita il contenuto dell'accumulatore sulla porta 0
033	000	000	Codice di dispositivo della porta 0
034	303	JMP	Salta incondizionatamente all'indirizzo di memoria contenuto nei due byte successivi
035	031	–	Byte di indirizzo LO
036	003	–	Byte di indirizzo HI

Passo 1

Premete il tasto RESET. Caricate 027 sul REGISTRO DATI e premete il tasto L.

Passo 2

Caricate in memoria il programma qui sopra partendo da HI 003 e LO = 027.

Passo 3

Caricate 027 nel REGISTRO DATI e premete il tasto L. Premete il tasto G. Cosa vedete sui LED della porta 0?

Noi abbiamo osservato che tutti gli indicatori a LED della porta 0 erano accesi. Non ci siamo accorti assolutamente che stavano lampeggiando.

ESPERIMENTO N. 4

Scopo

Lo scopo di questo esperimento è di caricare ed eseguire il "sesto programma' di questo capitolo. Il byte dati non è uguale a quello che abbiamo visto nel sesto programma.

Programma

Indirizzo di memoria LO	Byte di istruzione	Codice mnemonico	Descrizione
053	076	MVI A	Metti in modo immediato il byte dati seguente nell'accumulatore
054	300	300	Byte dati
055	062	STA	Memorizza il contenuto dell'accumulatore all'indirizzo di memoria contenuto nei due byte successivi.
056	061	—	Indirizzo di memoria LO
057	003	—	Indirizzo di memoria HI
060	166	HLT	Alt del microcomputer
061			Locazione di memoria dove memorizzare il contenuto dell'accumulatore.

Passo 1

Premete il tasto RESET. Caricate 053 nel REGISTRO DATI e premete il tasto L.

Passo 2

Caricate in memoria il programma partendo da HI = 003 e LO = 053.

Passo 3

Caricate 053 nel REGISTRO DATI e premete il tasto L. Premete il tasto G e infine il tasto RESET.

Passo 4

Caricate 061 nel REGISTRO DATI e premete il tasto L. Cosa vedete sugli indicatori a LED del REGISTRO DATI? Dovrebbe essere il byte di dati che avete memorizzato a HI = 003 e LO = 061.

DOMANDE RIEPILOGATIVE

Le seguenti domande vi aiuteranno a ripassare le istruzioni che avete usato in questo capitolo:

1. Spiegate cosa fa ciascuna di queste istruzioni

   a. 076
      133

   b. 303
      245
      003

   c. 323
      004

   d. 074

   e. 062
      344
      033

   f. 166

   g. 323
      000

   h. 000

2. Dite qual'è il codice ottale corrispondente a queste operazioni:

   a. Mettere in uscita il contenuto dell'accumulatore sulla porta 7

   b. Saltare all'indirizzo di memoria HI = 044 e LO = 123

   c. Memorizzare il contenuto dell'accumulatore nella locazione di memoria HI = 002 e LO = 070

   d. Mettere in modo immediato il byte dati contenente 222 nell'accumulatore.

   e. Incrementare il contenuto dell'accumulatore

   f. Fermare il microcomputer

   g. Nessuna operazione

RISPOSTE

1. a. Mette il byte dati 133 nell'accumulatore
   b. Salta all'indirizzo di memoria HI = 003 e LO = 245
   c. Mette in uscita il contenuto dell'accumulatore sulla porta 4
   d. Incrementa il contenuto dell'accumulatore
   e. Memorizza il contenuto dell'accumulatore nella locazione di memoria HI = 033 e LO = 344
   f. Fermare il microcomputer
   g. Mette in uscita il contenuto dell'accumulatore sulla porta 0
   h. Nessuna operazione

2. a. 323
   007
   b. 303
   123
   044
   c. 062
   070
   002
   d. 076
   222
   e. 074
   f. 166
   g. 000

Noi, come ci aspettavamo, abbiamo visto il numero 300.

CAPITOLO 6

REGISTRI E ISTRUZIONI RELATIVE AI REGISTRI

INTRODUZIONE

In questo capitolo imparerete alcune istruzioni di trasferimento tra registri e le impiegherete in alcuni programmi molto semplici sul microcomputer 8080A. Imparerete anche ad usare l'istruzione JNZ che vi permetterà di mettere dei *loop di delay (ritardo)* nei vostri programmi.

OBIETTIVI

Alla fine di questo capitolo sarete in grado di:

- Dare la definizione di registro.
- Elencare i sei registri non specializzati presenti nel microprocessor 8080A.
- Elencare quali di questi registri vengono usati come coppia di registri.
- Individuare una istruzione per il microprocssor 8080A nella lista di tutte le istruzioni possibili.
- Dire qual'è il codice ottale di un digit corrispondente ad ogni registro non specializzato.
- Spiegare l'istruzione di spostamento (MOVE) dati, MOV.
- Spiegare l'istruzione di spostamento immediato in un registro, MVI.
- Spiegare le istruzioni di incremento e decremento registro.
- Scrivere un programma molto semplice contenente un loop di delay.
- Scrivere un programma molto semplice che sposta dei dati da un registro all'altro.

COSA È UN REGISTRO?

Un *registro* è un circuito elettronico digitale di memorizzazione temporanea la cui capacità è di solito uguale ad una parola del computer. Nel microprocessor 8080A ogni singolo registro memorizza un solo byte, cioè otto bit congiunti. All'interno dell'8080A esiste un certo numero di registri, alcuni dei quali vengono usati per memorizzare dati e altri vengono invece usati esclusivamente per eseguire le istruzioni. Quindi possiamo suddividere i registri in due gruppi: quelli che potete indirizzare da programma e quelli che non potete indirizzare.
I registri indirizzabili da programma sono:

- Sei *registri non specializzati* a 8 bit indirizzati singolarmente o in coppie,

  registro B
  registro C
  registro D
  registro E
  registro H
  registro L

- l'*accumulatore* di 8 bit, detto anche *registro A*
- il *registro puntatore di STACK* di 16 bit (Stack Pointer)
- il *registro contatore di programma* di 16 bit (Program Counter)

Questi sono i soli registri con i quali potete scambiare direttamente informazioni mediante un programma appositamente scritto.

REGISTRI NON SPECIALIZZATI

I sei registri non specializzati - A, B, C, D, E, H e L - memorizzano temporaneamente un solo byte di informazione e, siccome si trovano tutti all'interno del microprocessore 8080A, lo scambio di informazioni tra due di essi è molto rapido, come è rapido anche lo scambio di informazioni tra uno di essi e l'accumulatore. Questi registri possono essere usati sia singolarmente che in coppia. I tre registri in coppia, quindi di 16 bit, sono:

- Il registro non specializzato di 16 bit formato dal registro A e dal registro B. Quando viene usato per indirizzare la memoria, il registro B corrisponde all'indirizzo HI e il registro C all'indirizzo LO.

- Il registro non specializzato di 16 bit formato dal registro D e dal registro E. Quando viene usato per indirizzare la memoria, il registro D corrisponde all'indirizzo HI e il registro E all'indirizzo LO.

- Il registro di indirizzo di memoria e quello non specializzato sono formati dal registro H e dal registro L. Quando viene usato per indirizzare la memoria, il registro H corrisponde all'indirizzo HI e il registro L all'indirizzo LO.

SET DI ISTRUZIONI DELL'8080A

Il set completo delle istruzioni dell'8080A è riportato nella pagina seguente nella forma suggerita da R. Baker e apparsa su BYTE, un periodico rivolto a coloro che hanno l'hobby del microcomputer. Nel capitolo 3 avete imparato che il codice operazione è il codice di otto bit corrispondente alla specifica operazione che il microprocessor 8080A può eseguire. Con otto bit di informazione, è possibile avere $2^8 = 256$ codici operazione diversi; questi codici sono quelli illustrati nelle pagine seguenti. I due digit della colonna più a siistra sono i primi due digit ottali dei tre digit ottali del codice operazione. Il rimanente digit è situato nella riga in alto e ripetuto altre tre volte all'interno della tabella.

A questo punto non vi chiediamo di imparare a memoria tutto il set di istruzioni. Il nostro scopo è solo quello di mostrarvi l'intero set di istruzioni a cui voi potrete fare riferimento ogni volta che imparate una nuova istruzione. Ad esempio nel capitolo 3 avete studiato le seguenti istruzioni:

000	NOP
062	STA**
074	INR A
076	MVI A*
166	HLT
303	JMP **
323	OUT *

Riuscite a trovarle nella tabella?

Osservate che il primo digit ottale - 0, 1, 2 o 3 - corrisponde alle quattro classi delle operazioni. Tutte le istruzioni MOV hanno un 1 nel primo digit ottale. Tutte le istruzioni aritmetico-logiche hanno un 2 nel primo digit ottale. Tutte le istruzioni che interrompono la normale sequenza di esecuzione - salti, chiamata di subroutines e rientri - hanno un 3 nel primo digit ottale. *Decodificare* una istruzione vuol dire individuare le relazioni che esistono tra ogni singolo bit e quello che essa fa: in ultima analisi è quanto viene eseguito dal *decodificatore di istruzioni* che si trova all'interno del microprocessor.

DECODIFICA DEI REGISTRI

Se studiate il set di istruzioni dell'8080A, potete osservare che ad ogni registro è assegnato un proprio codice ottale. Quindi:

Registro	Codice ottale
B	0
C	1
D	2
E	3
H	4
L	5
M	6
A	7

La lettera M si riferisce alla locazione di memoria indirizzata dal contenuto dei registri H e L. La lettera A si riferisce al registro accumulatore che abbiamo già discusso.

	0	1	2	3	4	5	6	7
00	NOP	LXI B **	STAX B	INX B	INR B	DCR B	MVI B *	RLC
01	—	DAD B	LDAX B	DCX B	INR C	DCR C	MVI C *	RRC
02	—	LXI D **	STX D	INX D	INR D	DCR D	MVI D *	RAL
03	—	DAD D	LDAX D	DCX D	INR E	DCR E	MVI E *	RAR
04	—	LXI H **	SHLD **	INX H	INR H	DCR H	MVI H *	DAA
05	—	DAD H	LHLD **	DCX H	INR L	DCR L	MVI L *	CMA
06	—	LXI SP **	STA **	INX SP	INR M	DCR M	MVI M *	STC
07	—	DAD SP	LDA **	DCX SP	INR A	DCR A	MVI A *	CMC

	0	1	2	3	4	5	6	7
10	MOV B,B	MOV B,C	MOV B,D	MOV B,E	MOV B,H	MOV B,L	MOV B,M	MOV B,A
11	MOV C,B	MOV C,C	MOV C,D	MOV C,E	MOV C,H	MOV C,L	MOV C,M	MOV C,A
12	MOV D,B	MOV D,C	MOV D,D	MOV D,E	MOV D,H	MOV D,L	MOV D,M	MOV D,A
13	MOV E,B	MOV E,C	MOV E,D	MOV E,E	MOV E,H	MOV E,L	MOV E,M	MOV E,A
14	MOV H,B	MOV H,C	MOV H,D	MOV H,E	MOV H,H	MOV H,L	MOV H,M	MOV H,A
15	MOV L,B	MOV L,C	MOV L,D	MOV L,E	MOV L,H	MOV L,L	MOV L,M	MOV L,A
16	MOV M,B	MOV M,C	MOV M,D	MOV M,E	MOV M,H	MOV M,L	HLT	MOV M,A
17	MOV A,B	MOV A,C	MOV A,D	MOV A,E	MOV A,H	MOV A,L	MOV A,M	MOV A,A

	0	1	2	3	4	5	6	7
20	ADD B	ADD C	ADD D	ADD E	ADD H	ADD L	ADD M	ADD A
21	ADC B	ADC C	ADC D	ADC E	ADC H	ADC L	ADC M	ADC A
22	SUB B	SUB C	SUB D	SUB E	SUB H	SUB L	SUB M	SUB A
23	SBB B	SBB C	SBB D	SBB E	SBB H	SBB L	SBB M	SBB A
24	ANA B	ANA C	ANA D	ANA E	ANA H	ANA L	ANA M	ANA A
25	XRA B	XRA C	XRA D	XRA E	XRA H	XRA L	XRA M	XRA A
26	ORA B	ORA C	ORA D	ORA E	ORA H	ORA L	ORA M	ORA A
27	CMP B	CMP C	CMP D	CMP E	CMP H	CMP L	CMP M	CMP A

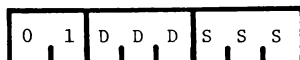
	0	1	2	3	4	5	6	7
30	RNZ	POP B	JNZ **	JMP **	CNZ **	PUSH B	ADI *	RST 0
31	RZ	RET	JZ **	—	CZ **	CALL **	ACI *	RST 1
32	RNC	POP D	JNC **	OUT *	CNC **	PUSH D	SUI *	RST 2
33	RC	—	JC **	IN *	CC **	—	SBI *	RST 3
34	RPO	POP H	JPO **	XTHL	CPO **	PUSH H	ANI *	RST 4
35	RPE	PCHL	JPE **	XCHG	CPE **	—	XRI *	RST 5
36	RP	POP PSW	JP **	DI	CP **	PUSH PSW	ORI *	RST 6
37	RM	SPHL	JM **	EI	CM **	—	CPI *	RST 7

Tabella dei codici di operazione sviluppata da R. Baker [*Byte 84* (Febbraio 1976)] Ogni * indica un byte di istruzione in più.

Per vedere come viene usato questo specifico codice ottale di un digit, bisogna fare riferimento alla tabella delle istruzioni; infatti potete notare che nelle istruzioni del gruppo da 20 a 27, IL TERZO DIGIT OTTALE CORRISPONDE AL REGISTRO INTERESSATO DALL'OPERAZIONE LOGICA O ARITMETICA. Nelle istruzione del set da 10 a 17 il terzo digit ottale corrisponde al registro contenente l'informazione che deve essere trasferita. Nelle istruzioni dei gruppi da 004 a 074, da 005 a 075 e da 007 a 076, il digit ottale di mezzo corrisponde al registro che è interessato all'operazione.

Adesso, o fate una copia del set completo di istruzioni dell'8080A, oppure trovate la regola che permette di ricostruire i codici istruzione; ma se volete diventare veramente esperti vi conviene fare tutte e due le cose.

TRASFERISCI DATI TRA DUE REGISTRI: MOV

Nel set dell'8080A ci sono 63 differenti istruzioni MOV, ed ognuna di esse ha il codice mnemonico del tipo MOV *D,S* dove *D*, = registro destinazione e *S* = registro sorgente. I codici istruzione vanno da 100 a 177, con la sola eccezione di 166 che è l'istruzione di HLT. Il formato degli 8 bit di una istruzione MOV è il seguente:

I valori di DDD o SSS sono dati dai tre bit che corrispondono al particolare codice ottale di un digit del registro, cioè:

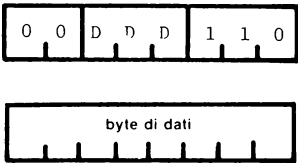
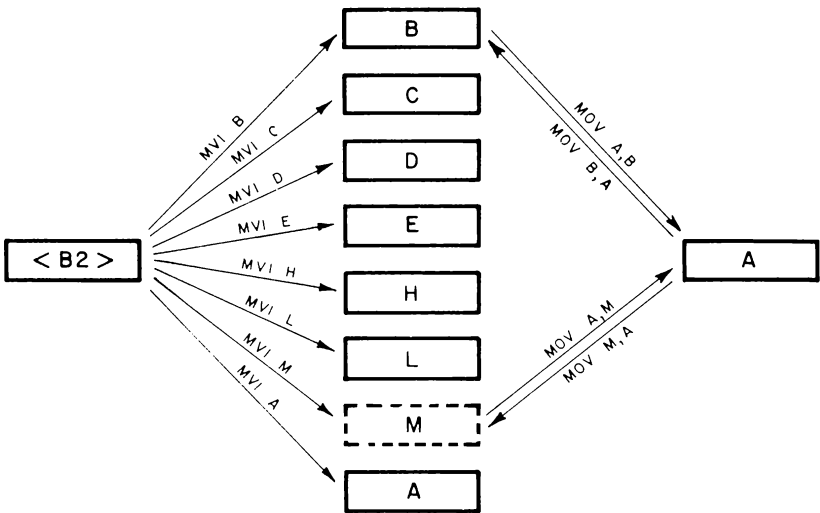
Registro	Codice ottale	Codice binario DDD o SSS
B	0	000
C	1	001
D	2	010
E	3	011
H	4	100
L	5	101
M	6	110
A	7	111

Il trasferimento di dati tra una locazione di memoria, M, ed un qualsiasi registro, abbisogna di ulteriori spiegazioni che non vanno date adesso. Tutti i trasferimenti di dati tra registri richiedono, per l'esecuzione, solo 5 cicli, cioè il tempo totale, per microcomputer con 750 kHz, è di 6,667 μs. Alcuni esempi di istruzioni apportate a questa classe sono qui sotto elencati:

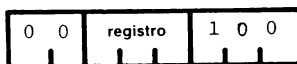
Muovi il contenuto del registro B nel registro C:	110	MOV C, B
Muvi il contenuto del registro C nell'accumulatore:	171	MOV A, C
Muovi il contenuto del registro E nel registro D:	123	MOV D, E
Mouvi il contenuto del registro H nell'accumulatore:	174	MOV A, H
Muovi il contenuto dell'accumulatre nel registro L:	157	MOV L, A
Muovi il contenuto dell'accumulatore nella memoria:	167	MOV M, A
Muovi il contenuto della memoria nell'accumulatore:	176	MOV A, M
Muovi il contenuto del registro L nel registro B:	105	MOV B, L
Muovi il contenuto del registro B nel registro B:	100	MOV B, B

Le istruzioni MOV A, B; MOV B, A; MOV A, M; MOV M, A vengono illustrate nella figura riportata più avanti, dove le frecce indicano la direzione del trasferimento dati.

TRASFERISCI DATI IN MODO IMMEDIATO VERSO UN REGISTRO: MVI

Come abbiamo già detto nel capitolo 3, il termine *in modo immediato* si riferisce al fatto che il byte di datiè contenuto all'interno della istruzione a più byte. Nella istruzione muovi-in-modo-immediato-nel-registro, il digit otale di mezzo indica la destinazione del byte di dati.

I valori di DDD sono quelli dati poco fa. Il codice mnemonico è MVI *D* e il numero di cicli richiesti è 7, che corrisponde ad un tempo di 9,333 μs, con l'eccezione della istruzione MVI M, la quale richiede 10 cicli e quindi 13,333 μs. Nella figura qui sotto abbiamo illustrato le otto diverse istruzioni muovi-in modo immediato-nel-registro. Il byte <B2> è il secondo byte di questa istruzione che ne contiene 2 e contiene l'informazione che viene trasferita dal programma al registro presclto.

Notate che *tutti i movimenti di registri e dati all'interno dell'8080A e del microcomputer MMD-1 sono in parallelo; cioè otto it di informazione vengono trasferiti contemporaneamente.* Per il trasferimento dati da e per il registro M sono richieste alcune condizioni speciali che discuteremo piu tardi.

INCREMENTA UN REGISTRO: INR

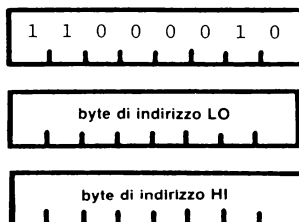
Incrementare un registro vuol dire aumentarne di 1 il contenuto. Questa istruzione, che è di un solo byte, ha la forma

e ha codice mnemonico INR r, dove r identifica il registro che deve essere incrementato. Sempre con l'eccezione della istruzione INR M, le altre richiedono 5 cicli, cioè 6,667 μs.

DECREMENTA UN REGISTRO: DCR

Decrementare un registro vuol dire diminuire di 1 il suo contenuto. Questa istruzione, che è di un solo byte, è del tutto simile a quella di incremento,

ed ha codice mnemonico DCR r. Con l'eccezione della DCR M, richiede 5 cicli, cioè 6,667 μs. Entrambe, sia l'istruzione di incremento che di decremento, usano un codice binario di tre bit per identificare il registro.

SALTA SE IL FLAG DI ZERO NON È UGUALE A ZERO: JNZ

Questa è la prima volta che vedete una *istruzione di salto condizionato* (Branch), cioè una istruzione che è soggetta ad una condizione. In questo caso, viene eseguito il salto all'indirizzo di memoria contenuto nel secondo e terzo byte dell'istruzione *se il flag di zero è al livello logico 0.* Per il momento non siamo ancora pronti a parlarvi di *flag,* ma per quello che ci serve adesso, possiamo dire che il salto avviene solo se il risultato di una operazione sui registri è zero, cioè 000. Se il registro diventa 000, l'istruzione viene ignorata e il programma salta i tre byte dell'istruzione e passa ad eseguire la successiva. L'istruzione JNZ è formata di tre byte

e richiede 10 cicli per l'esecuzione, cioè 13,333 μs. Questa istruzione è molto usata nei loop di delay, di cui viene fornito un esempio nel primo programma di questo capitolo.

PRIMO PROGRAMMA

Il primo programma che proverete è il seguente:

Indirizzo di memoria LO	Byte di istruzione	Codice mnemonico	Descrizione
062	006	MVI B	Metti in modo immediato il contenuto del byte dati seguente nel registro B
063	333	333	Byte dati
064	166	HLT	Alt del microcomputer

Quando eseguite questo programma, il byte dati 333 viene messo nel registro B all'interno del microprocessor e poi il microcomputer si ferma.

Come nel primo programma del capitolo 5, non potete accorgervi di cosa è successo, infatti nel programma non vi sono istruzioni di uscita e tutte le operazioni del microcomputer avvengono all'interno del microprocessor.

SECONDO PROGRAMA

Per fornire un'uscita al primo programma, dovete mettere i dati nell'accumulatore e poi fare un'uscita del suo contenuto verso un dispositivo esterno:

Indirizzo di memoria LO	Byte di istruzione	Codice mnemonico	Descrizione
065	006	MVI B	Metti in modo immediato il contenuto del byte dati seguente nel registro B
066	333	33	Byte dati
067	170	MOV A, B	Metti il contenuto del registro B nell'accumulatore
070	323	OUTMetti in uscita il contenuto dell'accumulatore sulla porta 1	
071	001	001	Codice di dispositivo della porta 1
072	166	HLT	Alt del microcomputer

L'istruzione MOV, A,B non modifica il contenuto del registro B dove rimane il dato 333. Infatti il trasferimento dati da un registro datiè un processo di copiatura e quindi il registro sorgente non viene nè cancellato nè modificato.

TERZO PROGRAMMA

In questo programma, trasferite un byte di dati da un registro all'altro e infine lo ponete in uscita prendendolo dall'accumulatore.

Indirizzo di memoria LO	Byte di istruzione	Codice mnemonico	Descrizione
073	006	MVI B	Metti in modo immediato il contenuto del byte dati seguente nel registro B
074	333	333	Byte dati
075	110	MOV C,B	Trasferisci il contenuto del registro B nel registro C
076	121	MOV D,C	Trasferisci il contenuto del registro C nel registro D
077	132	MOV E,D	Trasferisci il contenuto del registro D nel registro E
100	143	MOV H,E	Trasferisci il contenuto del registro E nel registro H
101	154	MOV L,H	Trasferisci il contenuto del registro H nel registro L
102	175	MOV A,L	Trasferisci il contenuto del registro L nel registro A, cioè nell'accumulatore
103	323	OUT	Poni in uscita il contenuto dell'accumulatore sulla porta 2
104	002	002	Codice di dispositivo della porta 2
105	166	HLT	Alt del microcomputer

Il byte dati viene caricato nel registro B e poi viene trasferito nel registro C,D,E,H,L e A seguendo l'ordine che abbiamo indicato. Alla fine dei trasferimenti, il dato è posto in uscita dall'accumulatore sulla porta 2 e dopo aver fatto queste operazioni, il microcomputer si ferma.

QUARTO PROGRAMMA

Questo programma è simile al terzo, con la sola differenza che, ad ogni passaggio del dato da un registro all'altro, viene eseguita una operazione aritmetica molto semplice:

Indirizzo di memoria LO	Byte di istruzione	Codice mnemonico	Descrizione
106	006	MVI B	Metti in modo immediato il byte dati seguente nel registro B
107	333	333	Byte dati
110	004	INR B	Incrementa il contenuto del registro B
111	110	MOV C,B	Trasferisci il contenuto del registro B nel registro C

112	014	INR C	Incrementa il contenuto del registro C
113	121	MOV D,C	Trasferisci il contenuto del registro C nel registro D
114	024	INR D	Incrementa il contenuto del registro D
115	132	MOV E,D	Trasferisci il contenuto del registro D nel registro E
116	035	DCR E	Decrementa il contenuto del registro E
117	143	MOV H,E	Trasferisci il contenuto del registro E nel registro H
120	045	DCR H	Decrementa il contenuto del registro H
121	154	MOV L,H	Trasferisci il contenuto del registro H nel registro L
122	055	DCR L	Decrementa il contenuto del registro L
123	175	MOV A,L	Trasferisci il contenuto del registro L nell'accumulatore
124	075	DCR A	Decrementa il contenuto dell'accumulatore
125	323	OUT	Poni in uscita il contenuto dell'accumulatore sulla porta 0
126	000	000	Codice di dispositivo della porta 0
127	166	HLT	Alt del microcomputer

Questo programma verrà discusso nell'esperimento n. 3. Tra il trasferimento dati da un registro all'altro, questi vengono incrementati o decrementati, quindi, visto che il dato era inizialmente pari a 333, alla fine sulla porta 0 ci sarà:

$$333 + 3 - 4 = 332$$

QUINTO PROGRAMMA

Questo è un programma molto interessante, contenente tra l'altro, un *loop di delay* che in pratica è una parte dell'intero programma, ripetutamente eseguita per una durata prefissata. Un loop di questo tipo può venire usato, per esempio, per rallentare una segnalazione di uscita che altrimenti cambierebbe troppo rapidamente. Ma prima di proseguire oltre vediamo il programma:

Indirizzo di memoria LO	Byte di istruzione	Codice mnemonico	Descrizione
130	076	MVI A	Metti in modo immediato il byte dati seguente nell'accumulatore
131	000	000	Byte dati

132	006	MVI B	Metti in modo immediato il byte dati seguente nel registro B
133	<B2>	<B2>	Byte di timing per il registro B
134	016	MVI C	Metti in modo immediato il byte dati seguente nel registro C
135	<B2>	<B2>	Byte di timing per il registro C
136	015	DCR C	Decrementa il contenuto del registro C
137	302	JNZ	Salta all'indirizzo di memoria contenuto nei due byte successivi di indirizzo, solo se il contenuto del registro C non è uguale a zero; altrimenti ignora questa istruzione.
140	136	--	Byte di indirizzo LO
141	003	--	Byte di indirizzo HI
142	005	DCR B	Decrementa il contenuto del registro B
143	302	JNZ	Salta all'indirizzo di memoria contenuto nei due successivi byte di indirizzo, solo se il contenuto del registro B non è uguale a zero; altrimenti ignora questa istruzione
144	134	--	Byte di indirizzo LO
145	003	--	Byte di indirizzo HI
146	323	OUT	Poni in uscita il contenuto dell'accumulatore sulla porta 2
147	002	002	Codice di dispositivo della porta 2
150	074	INR A	Incrementa il contenuto dell'accumulatore
151	303	JMP	Salta all'indirizzo di memoria contenuto nei due successivi byte di indirizzo
152	132	--	Byte di indirizzo LO
153	003	--	Byte di indirizzo HI

I due byte <B2>, che possono assumere un qualsiasi valore ottale compreso tra 000 e 377, sono dei byte di timing e determinano l'entità del tempo di ritardo.

Questo programma si comporta nel seguente modo:

- L'accumulatore e i registri B e C sono inizialmente caricati con dati o byte di timing. Qual'è la differenza tra un byte di dati e un byte di timing? In pratica non c'è nessuna differenza. La vera differenza stà solo nell'uso che si fa di questi dati.

- I byte di timing vengono usati nelle routines di delay, mentre i byte di dati sono, ad esempio, posti in uscita sulla porta 2.

- Il contenuto del registro C viene decrementato di 1. Se il risultato contenuto ancora nel registro C non è uguale a zero, il programma torna indietro con un salto all'indirizzo HI=003 e LO=136 dove il registro C viene decrementato ancora, e continua a decrementare il registro fino a che non diventa uguale a 000. A questo punto l'istruzione JNZ viene ignorata e il programma passa ad eseguire l'istruzione che si trova all'indirizzo HI = 003 e LO = 142.

- Quando il programma arriva all'indirizzo LO = 142 decrementa di 1 il registro B e, se questo diventa uguale a 000, la seguente istruzione JNZ viene ignorata, se il registro B non è uguale a zero il programma torna indietro con un salto all'indirizzo di memoria HI = 003 e LO = 134. A questo punto il registro C viene caricato col valore del byte di timing e il processo di decremento del registro C inizia nuovamente.

- Quando il programma arriva all'indirizzo di memoria LO = 146, il contenuto dell'accumulatore viene posto in uscita sulla porta 2, l'accumulatore incrementato di 1 e viene eseguita una istruzione JMP tornando indietro all'indirizzo di memoria HI = 003 e LO = 132 dove inizia nuovamente il processo di decremento dei registri B e C.

Questo programma è molto interessante perché contiene un loop all'interno di un loop che, a sua volta, è all'interno di un terzo loop. Il primo loop è formato dalle istruzioni DCR C e JNZ situate tra LO = 136 e LO = 141. Il secondo loop, che contiene il precedente, è compreso tra LO = 134 e LO = 145. In questo secondo loop, ogni volta che viene decrementato il registro B, il programma esegue un numero di passaggi nel primo loop pari al valore del byte di timing che si trova all'indirizzo LO = 135.

Il terzo loop si trova tra LO = 132 e LO = 153 e comprende l'incremento dell'accumulatore e l'uscita dell'accumulatore stesso sulla porta 2. Questo incremento avviene a intervalli regolari di tempo che si ripetono sulla base del tempo impiegato dal programma nell'eseguire il primo e secondo loop.

Proverete questo programma nell'Esperimento n. 5.

INTRODUZIONE AGLI ESPERIMENTI

I seguenti esperimenti vi mettono in grado di eseguire alcuni programmi molto semplici descritti in questo capitolo, programmi che vi faranno fare esperienza nell'uso delle istruzioni di trasferimento dati tra registri.

Gli esperimenti che eseguirete possono essere così riassunti:

Esperimento N.	Commento
1	Eseguire un programma che non prevede nessuna uscita del registro B
2	Eseguire un programma che pone in uscita il contenuto del registro B muovendolo prima nell'accumulatore e poi ponendolo in uscita direttamente dall'accumulatore.
3	Dimostrare l'uso delle istruzioni MOV, INR e DCR.
4	Provare che l'istruzione OUT esegue l'uscita di dati solo dall'accumulatore e non da un qualsiasi registro non specializzato.
5	Eseguire un programma molto semplice che contiene un loop di delay.
6	Eseguire un programma che contiene un loop di delay molto più lungo di quello dell'Esperimento n. 5.

ESPERIMENTO N. 1

Scopo

Lo scopo di questo esperimento è caricare ed eseguire il "primo programma" descritto in questo capitolo.

Programma

Indirizzo di memoria LO	Byte di istruzione	Codice mnemonico	Descrizione
062	006	MVI B	Metti in modo immediato il byte dati seguente nel registro B
063	333	333	Byte dati
064	166	HLT	Alt del microcomputer

Passo 1

Dopo aver dato tensione al microcomputer, premete il tasto RESET. Caricate 062 nel REGISTRO DATI e premete il tasto L.

Passo 2

Caricate in memoria il programma illustrato qui sopra partendo da HI = 003 e LO = 062.

Passo 3

Caricate 062 nel REGISTRO DATI e premete il tasto L. Premete il tasto G.

Passo 4

È successo qualcosa quando avete premuto il tasto G? Cosa?

A noi non è successo nulla. Il dato 333 è stato caricato nel registro B, ma non abbiamo la possibilità di verificarlo perchè tutte le operazioni sono avvenute all'interno del microprocessor. Abbiamo infatti bisogno di una istruzione di uscita di qualsiasi tipo per prendere il dato dal registro B e mandarlo ad un qualsiasi dispositivo che ci permetta o di visualizzarlo o di controllarne l'esistenza.

ESPERIMENTO N. 2

Scopo

Lo scopo di questo esperimento è di caricare ed eseguire il "secondo programma" descritto in questo capitolo

Programma

Indirizzo di memoria LO	Byte di istruzione	Codice mnemonico	Descrizione
065	006	MVI B	Metti in modo immediato il contenuto del byte dati seguente nel registro B
066	333	333	Byte dati
067	170	MOV A,B	Trasferisci il contenuto del registro B nell'accumulatore
070	323	OUT	Poni in uscita il contenuto dell'accumulatore sulla porta 1.
071	001	001	Codice di dispositivo della porta 1
072	166	HLT	Alt del microcomputer

Passo 1

Premere il tasto RESET. Caricate 065 nel REGISTRO DATI e premete il tasto L. Adesso caricate in memoria il programma illustrato qui sopra partendo da HI = 003 e LO = 065.

Passo 2

Caricate 065 nel REGISTRO DATI e premete il tasto L. Premete il tasto G. Cosa vedete sulla porta 1, cioè sul gruppo di LED più a sinistra?

Noi vediamo che i LED danno un'uscita pari a 333 indicando quindi che il byte di dati è stato trasferito dal registro B nell'accumulatore.

Passo 3

Premete ancora il tasto G. Non succede nulla? Perché?

Non succede nulla perché bisognava ripristinare il microcomputer prima di provare a rieseguire il programma. La causa di tutto è l'istruzione di Alt (HLT), 166.

ESPERIMENTO N. 3

Scopo

Lo scopo di questo esperimento è di caricare ed eseguire il "quarto programma" di questo capitolo. I byte di dati non sono uguali a quelli già descritti.

Programma

Indirizzo di memoria LO	Byte di istruzione	Codice mnemonico	Descrizione
106	006	MVI B	Metti in modo immediato il contenuto del byte dati seguente nel registro B
107	200	200	Byte dati
110	004	INR B	Incrementa il contenuto del registro B
111	110	MOV C,B	Trasferisci il contenuto del registro B nel registro C
112	014	INR C	Incrementa il contenuto del registro C
113	121	MOV D,C	Trasferisci il contenuto dal registro C nel registro D
114	024	INR D	Incrementa il contenuto del registro D
115	132	MOV E,D	Trasferisci il contenuto del registro D nel registro E
116	035	DCR E	Decrementa il contenuto del registro E
117	143	MOV H,E	Trasferisci il contenuto del registro E nel registro H
120	045	DCR H	Decrementa il contenuto del registro H
121	154	MOV L,H	Trasferisci il contenuto del registro H nel registro L
122	055	DCR L	Decrementa il contenuto del registro L
123	175	MOV A,L	Trasferisci il contenuto del registro L nell'accumulatore
124	075	DCR A	Decrementa il contenuto dell'accumulatore.
125	323	OUT	Poni in uscita il contenuto dell'accumulatore sulla porta 0
126	000	000	Codice di dispositivo della porta 0
127	166	HLT	Alt del microcomputer

Passo 1

Premete il tasto R e poi caricate in memoria il programma illustrato qui sopra partendo da HI = 003 e LO = 106. Per farlo dovreste impiegarci meno di un minuto e siccome è un programma abbastanza lungo, per favore, controllatelo tutto prima di eseguirlo.

Passo 2

Eseguite il programma posizionandovi all'indirizzo di partenza HI = 003 e LO = 106 e poi premendo il tasto G. Cosa succede?

Noi vediamo il valore 177 sugli indicatori a LED della porta 0 e questo è quanto ci aspettavamo. Infatti il contenuto del byte dati, 200, è stato incrementato tre volte e decrementato quattro volte, quindi 200 + 3 − 4 = 177 in codice ottale, naturalmente.

Questo programma è ottimo poiché vi permette di provare il comportamento di diverse istruzioni MOV, INR e DCR.

Passo 3

Cambiate adesso il contenuto del byte dati a LO = 107 con il valore 001 ed eseguite ancora il programma portandovi ad HI = 003 e LO = 106 e premendo il tasto G. Che numero vedete sugli indicatori a LED della porta 0?

Noi vediamo 000, che è quanto ci aspettavamo; infatti 001 + 3 − 4 = 000.

Passo 4

Cambiate il byte di dati a LO = 107 in 000 ed eseguite il programma un'ultima volta. Che risultato vedete sugli indicatori a LED della porta 0 e perché?

Noi vediamo 377 che è il risultato di 000 + 3 − 4 = 377. Ricordate quindi che se decrementate di 1 il valore 000 ottenete 377 e se incrementate 377 di 1, ottenete 000. Il passare oltre lo zero è indicato dal *carry flag;* ne discuteremo nel capitolo successivo.

ESPERIMENTO N. 4

Scopo

Scopo di questo esperimento è dimostrare che l'istruzione OUT pone in uscita dei dati solo dall'accumulatore.

Programma

Indirizzo di memoria LO	Byte di istruzione	Codice mnemonico	Descrizione
166	006	MVI B	Metti in modo immediato il byte dati nel registro B
167	000	000	Byte dati
170	016	MVI C	Metti in modo immediato il byte dati nel registro C
171	001	001	Byte dati
172	026	MVI D	Metti in modo immediato il byte dati nel registro D
173	002	002	Byte dati
174	036	MVI E	Metti in modo immediato il byte dati nel registro E
175	003	003	Byte dati
176	046	MVI H	Metti in modo immediato il byte dati nel registro H
177	004	004	Byte dati
200	056	MVI L	Metti in modo immediato il byte dati nel registro L
201	005	005	Byte dati
202	076	MVI A	Metti in modo immediato il byte dati nell'accumulatore
203	007	007	Byte dati
204	323	OUT	Poni in uscita il contenuto dell'accumulatore
205	000	000	Codice di dispositivo della porta 0
206	166	HLT	Alt del microcomputer

Passo 1

Premete il tasto RESET e caricate in memoria questo programma partendo da HI = 003 e LO = 166; prima di eseguirlo controllatelo tutto.

Passo 2

Caricate l'indirizzo di partenza, HI = 003 e LO = 166, nei registri di indirizzo di memoria e premete il tasto G per iniziare l'esecuzione. Che dato è posto in uscita sulla porta 0?

Noi vediamo, come uscita sulla porta 0, il valore 007.

Passo 3

Aggiungete 3 ad ogni byte di dati degli indirizzi LO 167, 171, 173, 175, 177, 201 e 203. I nuovi valori dei byte di dati saranno rispettivamente 003, 004, 005, 006, 007, 010 e 012.

Eseguite il programma una seconda volta. Che byte di dati vedete adesso sulla porta 0?

Noi vediamo sulla porta 0 il valore 012.

Passo 4

Riuscite ad identificare quale dei sette registri - B,C,D,E,H,L e accumulatore - viene posto in uscita dalla istruzione OUT?

Da questo esperimento si deduce che l'istruzione OUT riesce a mettere in uscita solo il contenuto dell'accumulatore.

Passo 5

Che procedimento dovete seguire per esaminare il contenuto dei sette registri - B,C,D,E,H,L e accumulatore -? Descrivetelo in dettaglio qui sotto.

Passo 6

Riuscite a capire quale registro riceve un dato in ingresso da un dispositivo d'ingresso usando la istruzione IN il cui codice è 333? Quale procedimento dovreste seguire per provare la vostra ipotesi?

ESPERIMENTO N. 5

Scopo

Lo scopo di questo esperimento è caricare ed eseguire un programma molto semplice contenente un loop di delay. Questo programma è più semplice del "quarto programma" di questo capitolo.

Programma

Indirizzo di memoria LO	Byte di istruzione	Codice mnemonico	Descrizione
154	015	DCR C	Decrementa il contenuto del registro C
155	302	JNZ	Ignora questa istruzione se il contenuto del registro C è 000, altrimenti salta all'indirizzo di memoria contenuto nei due byte successivi.
156	154	--	Byte di indirizzo LO
157	003	--	Byte di indirizzo HI
160	323	OUT	Poni in uscita il contenuto dell'accumulatore sulla porta 2
161	002	002	Codice di dispositivo della porta 2
162	074	INR A	Incrementa il contenuto dell'accumulatore
163	303	IMP	Salta all'indirizzo di memoria contenuto nei due byte di indirizzo immediatamente successivi.
164	154	--	Byte di indirizzo LO
165	003	--	Byte di indirizzo HI

Passo 1

Premete il tasto R e caricate in memoria il programma partendo da HI = 003 e LO = 154; dopo averlo caricato controllatelo. Poi, dato che avete ancora spazio nella memoria a lettura/scrittura, caricate la seguente istruzione di salto:

000	303	JMP	Salta al programma che si trova all'indirizzo di memoria contenuto nei prossimi due byte.
001	154	--	Byte di indirizzo LO
002	003	--	Byte di indirizzo HI

Passo 2

Premete il tasto R e poi G. L'esecuzione del programma inizia a HI = 003 e LO = 000, ma l'esecuzione passa immediatamente ad HI = 003 e LO = 154 perché incontra l'istruzione JMP.

Questo è un modo più semplice del precedente per far partire l'esecuzione dei vostri programmi. Infatti non dovete più caricare 154 nel REGISTRO DATI e premere il tasto L ogni volta che volete eseguirlo.

Che uscita vedete sulla porta 2?

A noi sembra che l'uscita sulla porta, visualizzata dagli otto indicatori a LED, venga continuamente incrementata.
Abbiamo detto "sembra" perché i bit meno significativi cambiano tanto rapidamente che non si riesce a vedere nessun tremolio mentre si può vedere bene il cambiamento degli altri indicatori a LED, quelli che rappresentano i bit più significativi. Quindi ci sembra che ci sia stato un incremento dei LED da 000 a 377 ripetuto più volte.

Passo 3

Approssimativamente quanto tempo passa per andare da 000 a 377?

Noi abbiamo eseguito la misura con un orologio da polso ed abbiamo contato ventitre cicli da 000 a 377 ogni trenta secondi cioè circa uno al secondo.

Passo 4

A cosa serve il "loop di delay" situato tra LO = 154 e LO = 157?

Questo loop viene eseguito 256 volte così il computer passa un certo tempo ad eseguire sempre le stesse istruzioni e quindi si produce un ritardo nella esecuzione del resto del programma.

Passo 5

Sostituite le istruzioni che si trovano tra LO = 154 e LO 157 con l'istruzione 177 cioè MOV A,A che trasferisce un dato dall'accumulatore nell'accumulatore, e quindi non produce cambiamenti nel contenuto di nessun registro. Cosa succede ora quando eseguite il programma?

Noi vediamo che sono accesi tutti i LED della porta 2, e quindi dobbiamo dedurre che i LED vengono incrementati ad una velocità troppo alta perché si possa percepire il cambiamento. Questo succede perché abbiamo tolto dal programma il loop di delay che rallentava l'esecuzione.

Passo 6

Rimettete al loro posto le istruzioni che avete tolto da LO = 154 a LO = 157. Cosa dovete cambiare nel programma perché i dati vengano posti in uscita sulla porta 0 invece che sulla porta 2? Fate questa modifica ed eseguite ancora il programma.

Abbiamo messo 000 al posto di 002 all'indirizzo LO = 161.

ESPERIMENTO N. 6

Scopo

Lo scopo di questo esperimento è caricare ed eseguire il "quinto programma" di questo capitolo

Programma

Indirizzo di memoria LO	Byte di istruzione	Codice mnemonico	Descrizione
130	076	MVI A	Metti in modo immediato il byte dati nell'accumulatore
131	000	000	Byte dati per l'accumulatore
132	006	MVI B	Metti in modo immediato il byte seguente nel registro B
133	000	000	Byte di timing per il registro B
134	016	MVI C	Metti in modo immediato il byte seguente nel registro C
135	000	000	Byte di timing per il registro C
136	015	DCR C	Decrementa il contenuto del registro C
137	302	JNZ	Ignora questa istruzione se il contenuto del registro C è 000; altrimenti salta all'indirizzo di memoria contenuto nei prossimi due byte.
140	136	––	Byte di indirizzo LO
141	003	––	Byte di indirizzo HI
142	005	DCR B	Decrementa il contenuto del registro B
143	302	JNZ	Ignora questa istruzione se il contenuto del registro B è 000; altrimenti salta all'indirizzo di memoria contenuto nei prossimi due byte.
144	134	––	Byte di indirizzo LO
145	003	––	Byte di indirizzo HI
146	323	OUT	Poni in uscita il contenuto dell'accumulatore sulla porta 2.

147	002	002	Codice di dispositivo della porta 2
150	074	INR A	Incrementa il contenuto dell'accumulatore
151	303	JMP	Salta all'indirizzo di memoria contenuto nei successivi due byte.
152	132	--	Byte di indirizzo LO
153	003	--	Byte di indirizzo HI

Passo 1

Premete il tasto R, caricate in memoria il programma pertendo da HI = 003 e LO = 130. Controllate tutto il programma.

Passo 2

Se avete ancora spazio disponibile nella memoria a lettura/scrittura caricate la seguente istruzione JMP:

000	303	JMP	Salta all'indirizzo di partenza del "quinto programma"
001	130	--	Byte di indirizzo LO
002	003	--	Byte di indirizzo HI

Passo 3

Per dare il via all'esecuzione di questo programma, potete scegliere tra due metodi:

- Caricare 130 nel REGISTRO DATI, premere il tasto L e poi premere G.
- Premere il tasto R e poi il tasto G, se avete caricato l'istruzione JMP all'indirizzo HI = 003 e LO = 000

Fate eseguire il programma. Che uscita vedete sulla porta 2?

Noi vediamo che i LED di questa porta vengono incrementati lentamente, una volta ogni 1,3 secondi circa.

Passo 4

Cambiate il contenuto del byte di timing a LO = 133 in 200 ed eseguite ancora il programma. Ogni quanto tempo si incrementano adesso i LED?

Noi vediamo che si incrementano ogni 0,67 secondi.

Passo 5

Cambiate il contenuto del byte di timing a LO = 133 in 100 ed eseguite il programma una terza volta. Ogni quanto tempo si incrementano adesso gli indicatori a LED?

Noi vediamo che gli indicatori a LED si incrementano ogni 0,33 secondi, cioè 3 volte ogni secondo.

Passo 6

I risultati che abbiamo ottenuto con questo esperimento possono essere così riassunti:

Contenuto del byte di timing a L=133	Intervallo tra due incrementi	Frequenza, incrementi/s.
100	0,33 s	3
200	0,67 s	1,5
000	1,33 s	0,75

Adesso spiegate con parole vostre perchè noi, e quindi anche voi, abbiamo ottenuto questi risultati. Questa è una domanda molto importante, perciò rispondete con esattezza. [Suggerimento: Che frequenza osservereste se il byte di timing fosse 040, 020 e 010?]

La risposta al suggerimento è che osservereste una frequenza rispettivamente di 6, 12 e 24 incrementi/secondo.

Passo 7

Eseguendo questo esperimento abbiamo avuto una certa difficoltà a leggere gli indicatori a LED mentre venivano incrementati, ma alla fine abbiamo trovato un modo per fermare il microcomputer al termine dell'esecuzione dell'istruzione in corso. Il sistema è molto semplice: premete il tasto R e tenetelo premuto fino a che non avete controllato o trascritto il numero ottale che appare come uscita sulla porta 2. Come lasciate andare il tasto RESET, il microcomputer ritorna all'indirizzo di memoria HI = 003 e LO = 000.

Mettete il byte di timing di LO = 133 a 040 e provate questo metodo per fermare l'esecuzione di un programma alla fine dell'esecuzione dell'istruzione in corso. Che frequenza di incremento vedete con questo valore del byte di timing?

Noi vediamo una frequenza di 6 incrementi/secondo.

DOMANDE RIEPILOGATIVE

Le seguenti domande vi aiuteranno a ripassare i registri non specializzati, i loop di delay e le istruzioni che avete usato in questo capitolo. Ripasserete anche il metodo di utilizzo della tabella delle istruzioni dell'8080A.

1. Cercate le seguenti istruzioni nella tabella delle istruzioni dell'8080A.

   a. MVI D *
   b. HLT
   c. NOP
   d. MOV A,A
   e. MOV B,B
   f. MVI A *
   g. MOV L,E

2. Quali istruzioni corrispondono ai seguenti codici ottali? Usate la tabella delle istruzioni dell'8080A.

   a. 076
   b. 333
   c. 302
   d. 303
   e. 311
   f. 005
   g. 004
   h. 006
   i. 154
   j. 145
   k. 123
   l. 132
   m. 176
   n. 177
   o. 002
   p. 377
   q. 307

3. Qual'è il codice ottale corrispondente alle seguenti istruzioni? Usate la tabella delle istruzioni dell'8080A.

   a. RNZ

   b. RZ

c. NOP

d. SBB C

e. ADD B

f. MOV A,A

g. RST 7

h. STAX B

i. DRC H

j. RLC

k. MOV E, D

l. MVI H *

m. JNZ *

n. POP B

o. PUSH B

p. LXI B **

q. PCHL

4. Elencate il codice ottale ad un digit, corrispondente ad ogni registro non specializzato.

5. Qual'è la differenza tra incrementare e decrementare?

6. Scrivete un programma che esegua le operazioni seguenti. Usate come indirizzo di partenza HI = 003 e LO = 000.

   Caricate 234 nel registro D. Trasferitelo nel registro H. Incrementate due volte il registro H. Trasferite il risultato nel registro B. Decrementate una volta questo dato. Traferite il risultato nell'accumulatore. Ponete il dato in uscita sulla porta 1. Fermate il microcomputer.

7. Scrivete un programma che contine solo un loop di delay.

RISPOSTE

1. A questa domanda non è prevista risposta.

2. a. MVI A *
   b. IN *
   c. INZ *
   d. JMP **
   e. RET
   f. DCR B
   g. INR B
   h. MVI B
   i. MOV L,H
   j. MOV H,L
   k. MOV D,E
   l. MOV E,D
   m. MOV A,M
   n. MOV A,A
   o. STAX B
   p. RST 7
   q. RST 0

3. a. 300

   b. 310

   c. 000

   d. 231

   e. 200

   f. 177

   g. 377

   h. 002

   i. 045

   j. 007

   k. 132

   l. 046

   m. 302

   n. 301

   o. 305

   p. 001

   q. 351

4. B = 0; C = 1; D = 2; E = 3; H = 4; L = 5; M = 6; Accumulatore = 7

5. L'incremento aumenta di 1 il contenuto del registro mentre il decremento lo diminuisce di 1.

6.

Indirizzo di memoria LO	Byte di istruzione	Codice mnemonico
000	026	MVI D
001	234	234
002	142	MOV H,D
003	044	INR H
004	044	INR H
005	104	MOV B,H
006	005	DCR B
007	170	MOV A,B
010	323	OUT
011	001	001
012	166	HLT

7.

Indirizzo di memoria LO	Byte di istruzione	Codice mnemonico
000	016	MVI C
001	000	000
002	015	DCR C
003	302	JNZ
004	000	––
005	003	––
006	166	HLT

CAPITOLO 7

PORTE LOGICHE E TABELLE DELLA VERITA'

INTRODUZIONE

Questo capitolo vi fornirà i concetti introduttivi alle porte logiche (logic gates) e alle tabelle della verità. Le porte logiche sono gli elementi base di tutti i dispositivi elettronici digitali.

OBIETTIVI

Al termine di questo capitolo sarete in grado di:

- Dare la definizione di dispositivo digitale.
- Dare la definizione di porta (gate)
- Dare la definizione di tabella della verità.
- Elencare i diversi usi delle porte.
- Scrivere correttamente le tabelle della verità per porte AND, NAND, NOR E OR.
- Spiegare la differenza tra un invertitore e un buffer.
- Scrivere correttamente la tabella della verità per una porta OR esclusivo (Exclusive-OR).
- Scrivere correttamente la tabella della verità per una porta AND-OR-INVERTITORE 2 - wide 2 input
- Spiegare cosa significa il cerchietto all'ingresso o all'uscita di un dispositivo digitale che potrebbe essere, per esempio, una porta.
- Dimostrare come ogni porta può essere trasformata in una qualsiasi altra con l'ausilio di uno o più invertitori.

COSA È UN DISPOSITIVO DIGITALE?

Tenendo conto degli obiettivi che questa trattazione si propone, un *dispositivo digitale* può essere definito come un dispositivo di qualsiasi tipo che opera o manipola informazioni *binarie*, o a due stati. Come ricorderete ogni codifica binaria può essere rappresentata da un qualsiasi tipo di dispositivo bistabile come una luce accesa o spenta, un interruttore aperto o chiuso, una scheda meccanografica perforata o non perforata, un nucleo magnetico o una parte di nastro o disco magnetizzati "nord' o 'sud'; due livelli di tensione diversi, due livelli di corrente diversi, due frequenze diverse; oppure dei simboli astratti 0 (off) e 1 (on).

Nell'elettronica digitale, ci interessano in particolare i dispositivi elettronici digitali che trattano informazioni digitali nella forma di livelli di tensione, +5 V per lo stato logico 1 e potenziale di massa per lo stato logico 0. Grazie ai recenti progressi dell'industria elettronica è oggi possibile costruire dispositivi che possono cambiare stato molto rapidamente, in tempi più brevi di 2 ns (0,000000002).
D'altro canto un grande numero di questi tipi di dispositivi può essere acquistato a prezzi molto bassi. Due considerazioni importanti - velocità e basso costo - si sono combinate in modo tale da rendere l'elettronica digitale una componente importante per lo sviluppo della società soprattutto nell'area delle comunicazioni, dei controlli e della elaborazione dati.

Tra i dispositivi che incontrerete leggendo questa trattazione vi sono i seguenti:

porte
flip-flops
contatori
decodificatori
buffers
registri di shift
memorie
multiplexers
demultiplexer
unità artimetico/logiche
priority encoders
latches

Inizieremo con l'introduzione del più semplice dispositivo digitale, *cioè* la porta logica (logic gate).

COSA É UNA PORTA?

Una *porta* è un circuito con due o più ingressi e un'uscita; l'informazione digitale che si ottiene all'uscita dipende dalla combinazione di informazioni digitali che si presentano agli ingressi. Oggi la parola "porta" ha più significati nel campo dell'elettronica; in questa trattazione faremo riferimento solo alle porte come dispositivi digitali.

I quattro tipi più comuni di porte sono AND, OR, NAND e NOR. Una porta un po' più complicata che può essere costruita con dei NAND o dei NOR è l'OR-esclusivo. Però, prima di parlare di queste porte, discuteremo il concetto di tabella della verità.

COSA È UNA TABELLA DELLA VERITA'?

Una *tabella della verità* è una rappresentazione tabellare che mostra le correlazioni tra tutti i livelli logici in uscita da un circuito digitale rispetto a tutte le possibili combinazioni di livelli logici in ingresso e quindi definisce completamente le funzioni del circuito. Parlando di "livelli logici" in ingresso e in uscita abbiamo fatto riferimento agli stati logici 0 e 1, che in molti dispositivi elettronici digitali sono rappresentati da due livelli di tensione, rispettivamente 0 V e +5 V. Adesso possiamo ridefinere una tabella della verità nel seguente modo: una tabella della verità è una rappresentazione tabellare che mostra le correlazioni tra gli stati 0 logico ed 1 logico in uscita da un circuito digitale rispetto a tutte le possibili combinazioni di stati in ingresso, definendo completamente le funzioni del circuito.

La figura seguente vi mostra un ipotetico circuito digitale con tre ingressi o input (A,B, e C) e quattro uscite o output (Q1, Q2, Q3 e Q4). In base a quello che abbiamo appena detto, lo scopo di una tabella della verità è quello di difinire completamente le relazioni esistenti tra gli ingressi e le uscite. Non è un problema complicato, poiché gli ingressi e le uscite possono assumere solo i valori 0 e 1. Allora, poiché esistono solo due possibili stati per ogni ingresso, e siccome ci sono tre ingressi, abbiamo in totale $2^3 = 8$ diversi possibili combinazioni in ingresso , che abbiamo elencato nella figura sotto riportata.

Ingressi			Uscite			
C	B	A	Q1	Q2	Q3	Q4
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

Nella parte sinistra della tabella della verità, sono riportati gli otto diversi gruppi di tre possibili ingressi. A questo punto sorge spontanea una domanda; come è possibile determinare gli stati logici delle quattro uscite, da Q1 a Q4? La risposta è: gli stati logici delle quattro uscite dipendono dalla struttura del particolare circuito logico. Non si può dire altro con le tabelle della verità fino a che non sia noto lo specifico circuito cui si vuol fare riferimento. Nella precedente tabella della verità, esistono $2^{32} = 4.294.967.296$ differenti combinazioni di valori di uscita rispetto a otto diversi ingressi. (Se non avessimo avuto a disposizione una calcolatrice digitale tascabile, non avremmo certo potuto fare i calcoli così facilmente, e la calcolatrice è solo uno dei moderni ritrovati della tecnologia dell'elettronica digitale).
In conclusione: *non si può completare una tabella dellà verità per un certo gruppo di ingressi, fino a che non è noto lo schema del circuito interessato, o fino a che non è noto il nome del circuito interessato.*

PERCHÈ SI USANO LE TABELLE DELLA VERITA'

Le tabelle della verità vengono usate poiché permettono di dare, in modo semplice, una adeguata rappresentazione dei circuiti digitali. È molto più facile scrivere una tabella della verità che spiegare a parole il funzionamento di un circuito digitale. Il vantaggio di una tabella della verità rispetto ad una descrizione a parole, consite nel poter dare contemporaneamente i valori di più ingressi e più uscite in modo chiaro. Noi usiamo le tabelle per rappresentare i circuiti digitali come i tecnici elettronici usano gli schemi dei circuiti per rappresentare i dispositivi elettronici, gli architetti usano le carte cianografiche per rappresentare le strutture architettoniche ed i chimici usano i disegni bi-dimensionali per rappresentare i legami chimici. In altre parole, le tabelle della verità, danno una rappresentazione del comportamento dei circuiti digitali.
Ricordate il detto: "Una fotografia vale mille parole?" Ecco, una tabella della verità ne vale venti.

UTILIZZO DELLE PORTE

Prima di passare a discutere i simboli e le caratteristiche di ogni singola porta, precisiamo che le porte - le funzioni elementari dell'elettronica digitale - possono essere usate in diversi modi, però indipendentemente da come una porta è usata, *la tabella della verità fondamentale per quella porta è sempre valida.* Allora ne consegue che una tabella della verità è una caratteristica base di una specifica porta, e non viene alterata in nessun caso dall'uso che si fa di quella porta.

Alcuni modi di usare le porte sono:

- Nei "gating circuits" (circuiti di abilitazione)

  Un *gating circuit* è un circuito in cui una porta è usata per controllare il passaggio di un segnale digitale, che può essere un singolo impulso, un gruppo di impulsi o un treno di impulsi ad una frequenza conosciuta o no.

- Nei circuiti logici

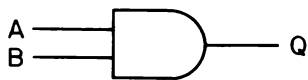
  Un *circuito logico* è un circuito elettronico che stabilisce una relazione tra ingressi e uscite, corrispondente ad una funzione logica secondo *l'algebra Booleana.*

- Come componenti di elementi di memoria

- Una *memoria* è un qualsiasi dispositivo che può memorizzare gli stati logici 0 e 1 in modo tale che poi si possa accedere e leggere un singolo bit od un gruppo di bit. Gli elementi di memoria possono essere costruiti con gruppi di porte.

SIMBOLI DELLE PORTE

Le rappresentazioni simboliche delle porte sono diventate standard per merito della industria elettronica e noi useremo questi simboli standard sia nella trattazione che negli esperimenti presentati. Tra i simboli da ricordare ci sono:

Porta AND a 2 - ingressi

Porta AND a 3 - ingressi

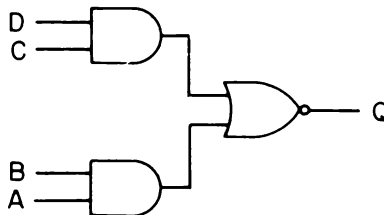
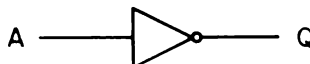
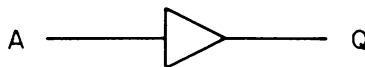
Porta NAND a 2 - ingressi

Porta NAND a 3 - ingressi

Porta OR a 2 - ingressi

Porta NOR a 2 - ingressi

Porta OR-esclusivo

che può essere costituita con porte NAND o NOR a 2 ingressi e la porta AND-OR-INVERTITORE

La porta AND-OR-INVERTITORE, mostrata qui sopra, è una porta a *2-Wide 2 Input* (letteralmente: di larghezza doppia e con due ingressi); tale dispositivo è funzionalmente costituito da due porte AND (da cui la denominazione di larghezza doppia), ciascuna delle quali a 2 ingressi, da una porta OR e da un invertitore all'uscita.
Oltre alle porte sin qui viste, ci sono anche simboli per dispositivi digitali con un solo ingresso ed una sola uscita, dispositivi come *l'invertitore,*

ed il *Buffer,* detto anche *Driver.*

Dopo aver visto i simboli, parleremo delle funzioni di ciascuna di queste porte e accompagneremo la trattazione con le tabelle della verità relative ad ognuna di esse.

PORTA AND

Il comportamento di una porta AND a 2 ingressi può essere espresso come segue: se l'ingresso A è uguale ad 1 logico e l'ingresso B è uguale ad 1 logico, allora l'uscita Q è uguale ad 1; in tutti gli altri casi l'uscita Q è uguale a 0 logico.

Ripetiamo il simbolo di questa porta, accompagnandolo con la sua tabella della verità.

Ingressi		Uscita
B	A	Q
0	0	0
0	1	0
1	0	0
1	1	1

L'"Oxford English Dictionary" definisce la parola *unico* come: "è il solo del suo tipo; non ha simili o uguali; non può essere confrontato con altri; uno e nessun altro; singolo, solo o solitario". Nel senso dato alla parola dal dizionario, esiste uno stato unico dell'uscita per la porta AND a due ingressi ed è lo stato dell'uscita corrispondente agli ingressi A e B entrambi allo stato logico 1. Ripetiamo questa osservazione perché sarà importante quando tenterete di ricordare la tabella della verità per le porte AND: *Si ha uno stato di uscita unico, 1 logico, quando tutti gli ingressi di una porta AND sono al livello logico 1.*

Il comportamento di una porta AND a 3 ingressi può essere enunciato come segue: se l'ingresso *A* e l'ingresso *B* e l'ingresso *C* sono 1, allora l'uscita Q è 1; in tutti gli altri casi, l'uscita Q è 0. Il simbolo di questa porta e la sua tabella della verità sono i seguentii:

Ingressi			Uscita
C	B	A	Q
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Possiamo allora definire una *porta AND* come un circuito binario con 2 o più ingressi e un'unica uscita, che si trova allo stato logico 1 solo se tutti gli ingressi sono ad 1, e nel quale l'uscita è allo 0 logico se almeno uno degli ingressi è 0.

PORTA NAND

Una *porta NAND* può essere definita come un circuito binario con 2 o più ingressi e un'unica uscita, nel quale l'uscita è 0 quando tutti gli ingressi sono 1, e nel quale l'uscita è 1 se almeno uno degli ingressi è 0. La porta NAND è strettamente legata alla porta AND; la sola differenza è data dal fatto che l'uscita della porta NAND è il complemento dell'uscita della porta AND equivalente. Il significato del termine complemento sarà più chiaro tra un attimo.

Il simbolo e la tabella della verità per una porta NAND a 2 ingressi sono i seguenti:

Ingressi		Uscita
B	A	Q
0	0	1
0	1	1
1	0	1
1	1	0

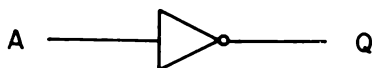
Confrontate questa tabella della verità con quella della porta AND a 2 ingressi. Sono assolutamente identiche escluse le uscite: Se è 1 per la porta AND, è 0 per la porta NAND e quando è 0 per la porta AND, è 1 per la porta NAND. Si dice in questo caso che gli stati logici di una porta sono i *complementi dell'altra.*
Lo stato logico 1 è il complemento dello stato logico 0 e 0 è il complemento di 1.

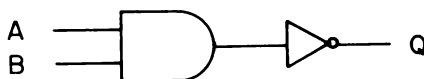
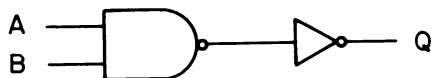
Il simbolo e la tabella della verità per una porta NAND a 3 ingressi sono i seguenti:

Ingressi			Uscita
C	B	A	Q
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

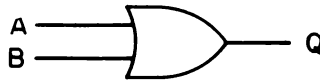
INVERTITORE

L'*invertitore* o *porta NOT* è definito come un circuito binario con un ingresso e un'uscita, nel quale l'uscita è 1 quando l'ingresso è 0 e l'uscita è 0 quando l'ingresso è 1. In effetti l'uso del termine "porta NOT" non è appropriato; infatti un invertitore non è una porta, perché ha un solo ingresso.

Il simbolo e la tabella della verità per un invertitore sono i seguenti:

Ingresso A	Uscita Q
0	1
1	0

Osservate che i 2 ingressi della porta NAND possono essere sostituiti dai 2 ingressi della porta AND con un invertitore posto all'uscita della porta AND.

Osservate inoltre che i 2 ingressi della porta AND possono essere sotituiti dai 2 ingressi della porta NAND con un invertitore posto all'uscita della porta NAND.

PORTA OR

Una *porta OR* è un circuito binario con 2 o più ingressi e una sola uscita, nel quale l'uscita è 0 solo quando tutti gli ingressi sono 0, e nel quale l'uscita è 1 se almeno uno degli ingressi è 1. Il comportamento di una porta OR a 2 ingressi può essere enunciato come segue: se l'ingresso A è allo stato logico 1 *o* l'ingresso B è allo stato logico 1, allora l'uscita Q è allo stato logico 1; altrimenti l'uscita è allo stato logico 0. Quanto detto contempla anche il caso che sia l'ingresso A che B siano allo stato logico 1.

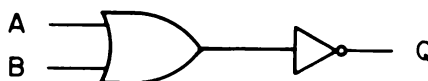
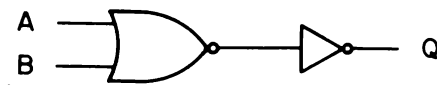
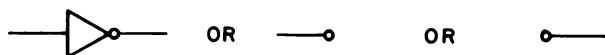
Il simbolo e la tabella della verità per una porta OR a 2 ingressi sono i seguenti:

Ingressi		Uscita
B	A	Q
0	0	0
0	1	1
1	0	1
1	1	1

In questo caso, lo stato unico si ha quando entrambi gli ingressi A e B sono a 0. Questa osservazione diventerà importante quando cercherete di ricordare la tabella della verità per una porta OR: *Si ha lo stato di uscita unico, stato logico 0, quando tutti gli ingressi di una porta OR sono allo stato logico 0.*

PORTA NOR

Una *porta NOR* è un circuito binario con 2 o più ingressi ed una sola uscita, nel quale l'uscita è allo stato logico 1 solo quando tutti gli ingressi sono allo stato logico 0, e nel quale l'uscita è allo stato logico 0 se almeno uno degli ingressi è allo stato logico 1. Il simbolo e la tabella della verità per una porta OR a 2 ingressi sono i seguenti:

Ingressi		Uscita
B	A	Q
0	0	1
0	1	0
1	0	0
1	1	0

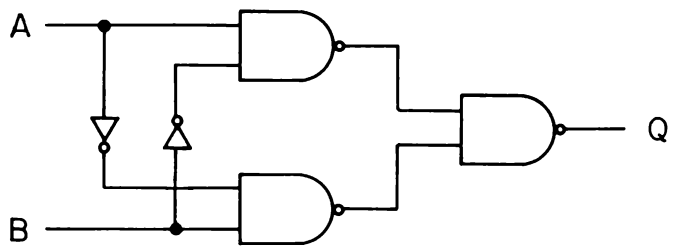
Osservate il cerchietto alla destra del simbolo della porta OR; avete visto questo cerchio precedentemente, nel simbolo della porta NAND. In entrambi i casi il simbolo rappresenta la negazione, l'inversione o il complemento. In altre parole, in una porta NOR si trattano gli ingressi come una porta OR, e poi si inverte l'uscita. Una porta NOR a 2 ingressi può essere costituita da una porta OR e un singolo invertitore,

e una porta OR a 2 ingressi può essere costituita da una porta NOR a 2 ingressi e un invertitore esterno:

Osserviamo bene che l'inversione, o complementazione, può essere rappresentata sia col simbolo dell'invertitore che con un picccolo cerchio messo prima o dopo una porta o altro simbolo di dispositvo digitale:

Nel riepilogo di questo capitolo, chiederemo di dimostrare come un tipo di porta può essere convertito in un altro usando gli invertitori.

PORTA OR-ESCLUSIVO

Per definire la porta *OR-Esclusivo* è meglio usare il suo simbolo e la sua tabella della verità,

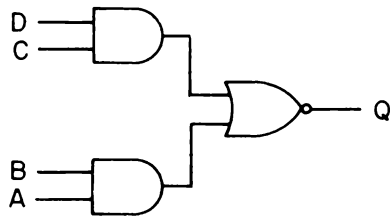
Ingressi		Uscita
B	A	Q
0	0	0
0	1	1
1	0	1
1	1	0

La porta OR-Esclusivo è strettamente correlata con la porta OR, che è detta anche porta OR-*Inclusivo.* La sola differenza tra le due si ha quando entrambi gli ingressi A e B sono allo stato 1: per una porta OR, l'uscita è 1, mentre per una porta OR-Esclusivo l'uscita è 0. Mentre la porta OR a più ingressi è usata di rado, la porta OR-Esclusivo è usata solo a 2 ingressi come abbiamo visto prima.

Le porte OR-Esclusivo sono abbastanza comuni e possono essere acquistate singolarmente o nella forma di circuiti integrati che ne contengono quattro. Una singola porta OR-Esclusivo può essere costruita come mostrato di seguito con tre porte NAND a 2 ingressi e un paio di invertitori:

Se ricordate le tabelle della verità della porta NAND e dell'invertitore, potete facilmente dimostrare che questo circuito ha la stessa tabella della verità della porta OR-esclusivo. Vi chiederemo di farlo nel riepilogo di questo capitolo.

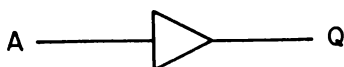
PORTA AND-OR-INVERTITORE

La *porta AND-OR-INVERTITORE* è un circuito composto di porte ed è molto usato nei computer. Non è una porta base, infatti è formata da 2 porte AND e da una porta NOR. Questo tipo di porta è reperibile in coppia sui circuiti integrati. Quando usate una porta AND-OR-INVERTITORE dovete specificare quante porte AND ci sono e quanti ingressi ha ognuna di esse. Quando dite *porta AND-OR-INVERTITORE a 3 wide e 2 input,* intendete dire che ci sono 3 porte AND ognuna a 2 ingressi.

Il simbolo e la tabella della verità per una porta AND-OR-INVERTITORE a 2 wide e 2 ingressi sono i seguenti:

Ingressi				Uscita
D	C	B	A	Q
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

BUFFER/DRIVER

Una *porta buffer o driver* è un circuito digitale che, connesso ad un altro circuito binario, ne aumenta la capacità in termini di potenza o corrente. È un circuito binario con un ingresso ed un'uscita, nel quale l'uscita è 0 quando l'ingresso è 0, e nel quale l'uscita è 1 quando l'ingresso è 1. Noi preferiamo il termine *buffer* piuttosto che porta buffer, perché c'é un solo ingresso.

Il simbolo e la tabella della verità per un buffer non invertente o driver sono i seguenti

Ingresso A	Uscita Q
0	0
1	1

In elettronica, la potenza è data dalla corrente moltiplicata per la tensione, quindi è facile capire come può aumentare la potenza anche se la tensione non cambia.
Importante: *La tensione in uscita di un buffer o driver è normalmente uguale a +5 V o uguale al potenziale di massa.*
In qualche caso, tuttavia, la tensione in uscita è molto più grande ed è usata per pilotare una valvola NIXIE®, lampade a filamento incandescente e altri dispositivi che richiedono una tensione superiore a +5 V. In questi casi, *il circuito buffer/driver non deve mai essere collegato agli ingressi degli altri circuiti integrati.*

"GATING CIRCUITS" PIU' COMPLESSI

Gli elementi base di ogni circuito digitale sono le porte oppure le porte con l'invertitore. Potete costruire un IBM 370 con porte ed invertitori avendo una sufficiente quantità di:

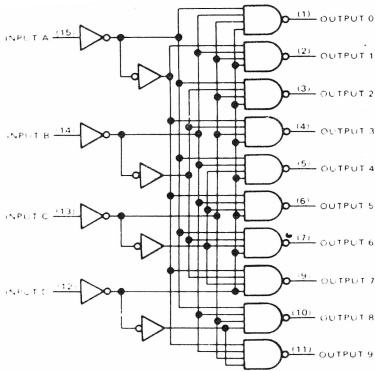
- Porte NAND o
- Porte NOR o
- Porte AND e invertitori o
- Porte OR e invertitori

In altre parole, supponendo che abbiate un numero sufficiente di porte NAND, comprendenti porte a 2,3,4,8 etc. ingressi, potreste costruire qualsiasi circuito digitale se avete anche abbastanza cavo elettrico e pazienza. Potete farlo anche se avete un numero sufficiente di porte AND e di invertitori.

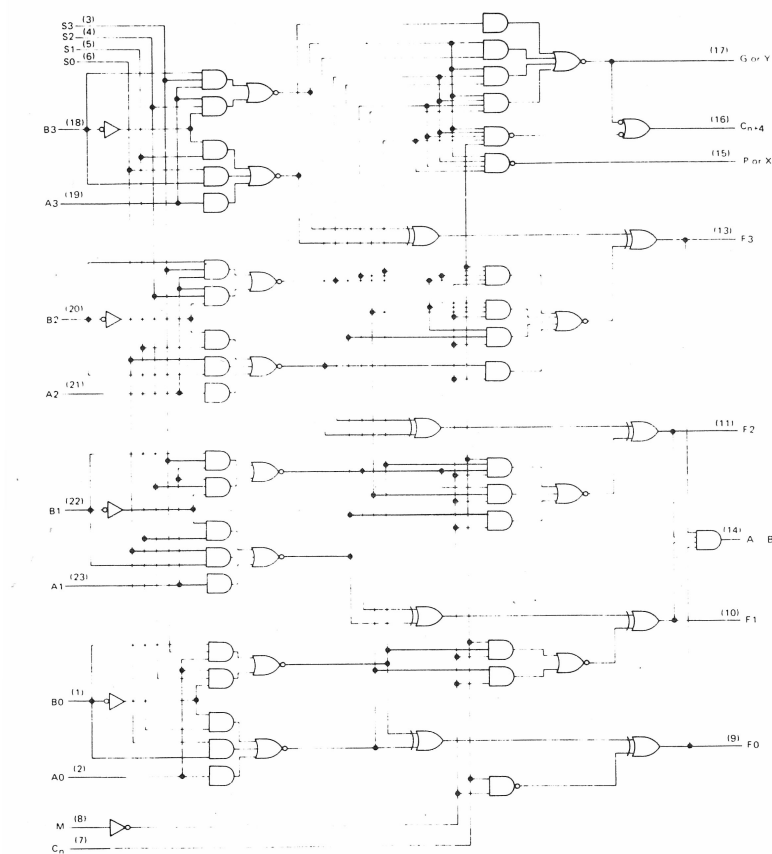
Per fortuna non dovete costruire un IBM 370, e nemmeno un *decodificatore,* partendo da porte e invertitori.

I costruttori di dispositivi digitali elettronici hanno già fatto il lavoro per voi. In realtà sono disponibili, ad un costo veramente basso, dei circuiti integrati che contengono gating circuits abbastanza complessi come *decodificatori, full-adders, multiplexers, generatori di funzioni, unità aritmetico/logiche, priority encoder.* Come esempio di circuito a porte abbastanza complesso, guardate la tabella della verità e il circuito di un *decodificatore da BCD a decimale*

Ingressi				Uscite									
D	C	B	A	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1	1	1
0	1	1	0	1	1	1	1	1	1	0	1	1	1
0	1	1	1	1	1	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0
1	0	1	0	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1

functional block diagram

Per costruire un circuito di questo tipo sono occorsi dieci porte NAND a 4 ingressi, quattro invertitori, e 4 buffer/invertitori. Tutto questo vi è dato in un singolo circuito integrato. Lo schema che avete visto è un 7442 riprodotto per concessione della Texas Instruments Incorporated.

Un esempio di un circuito a porte ancora più complesso è quello del 74181 arithmetic logic unit/function generator, il cui schema è riprodotto di seguito

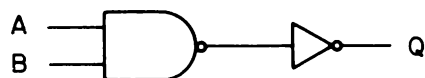
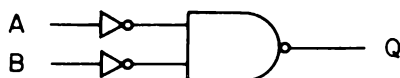
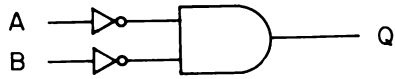
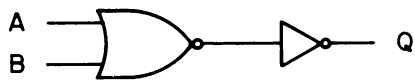
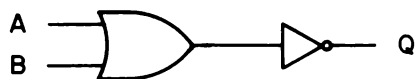
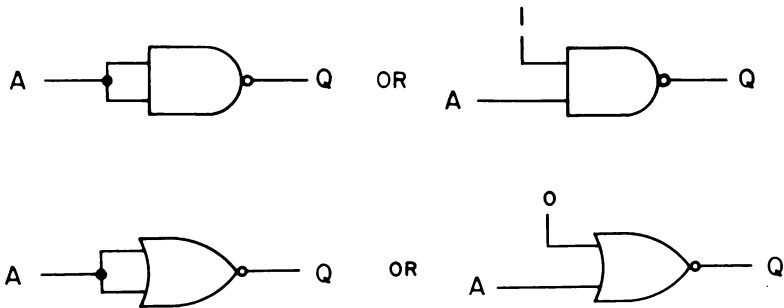
functional block diagram

Se studiate attentamente questo circuito potete osservare le porte AND-OR-INVERTITORE, le porte OR-esclusivo, gli invertitori, le porte NAND a 4 ingressi e le porte AND a 4 ingressi. Anche questo circuito è disponibile in un solo circuito integrato, il 74181 e anche questo schema è riprodotto per gentile concessione della Texas Instruments Incorporated.

Vi abbiamo proposto questi complicati circuiti col solo scopo di attirare l'attenzione su un punto importante. *Per coloro che iniziano a studiarla, l'elettronica si fa sempre più facile, invece che più complessa, e* questa è la conseguenza del fatto che le industrie costruiscono sempre più circuiteria in un solo circuito integrato. *In futuro avrete a che fare con sempre più circuiteria in un chip e sempre meno fili all'esterno.*

DOMANDE RIEPILOGATIVE

Le seguenti domande hanno lo scopo di aiutarvi a ripassare le porte logiche e le tabelle della verità. Alcune di esse sono più difficili delle domande che avete trovato alla fine di ogni capitolo precedente.

1. Avendo a disposizione un numero sufficiente di invertitori, è possibile convertire una qualsiasi porta base in un'altra porta base. Per ognuno degli schemi riportati di seguito, scrivete la tabella della verità nella quale A e B sono gli ingressi e Q è l'uscita. Dopo aver fatto questo scrivete il nome della porta corrispondente alla tabella fatta.

f.
A
B
Q
g.
A
B
Q
h.
A
B
Q
i.
A
B
Q
j.
A
B
Q
k.
A
B
Q

2. È possibile convertire sia una porta NAND che una porta NOR in un invertitore. Negli schemi sottostanti, mostriamo come sia possibile farlo per le porte a 2 ingressi.

Mostrate i due modi in cui le seguenti porte possono essere trasformate in invertitori:

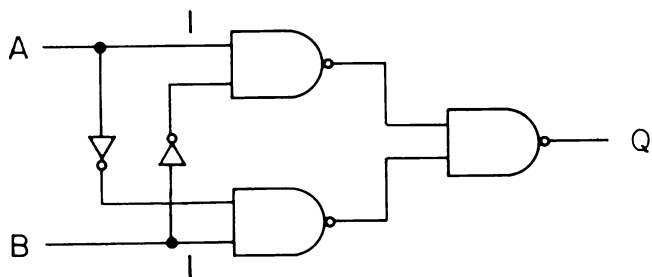
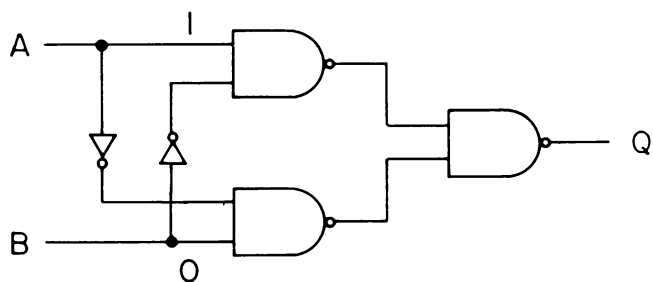
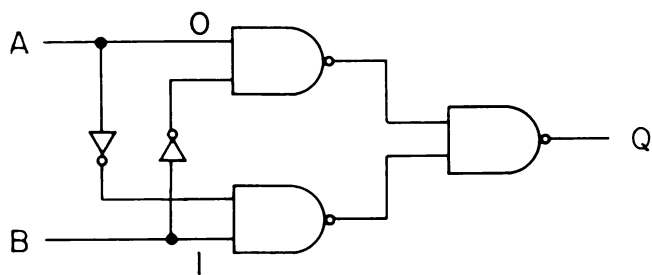
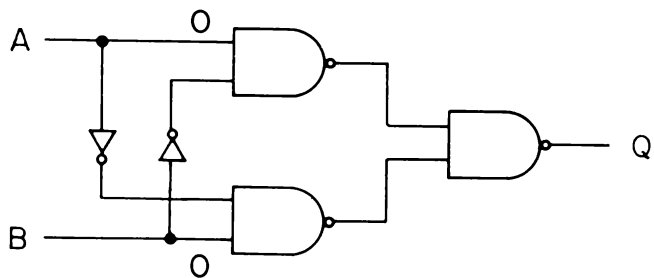
a. Porta NAND A 3-ingressi

b. Porta NOR a 3-ingressi

c. Porta NAND a 4-ingressi

d. Porta AND a 4-ingressi

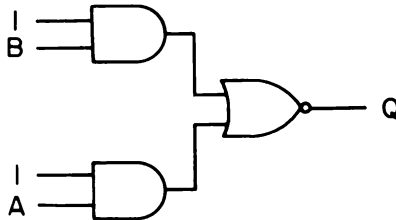
3. Qui di seguito sono mostrati quattro schemi identici per una porta OR-Esclusivo che è costituita con 3 porte NAND a 2 ingressi e un paio di invertitori. Sono possibili quattro tipi di ingressi: A=0 e B=0; A=1 e B=0; A=0 e B=1; A=1 e B=1. Dite per ognuno di essi quali sono gli stati logici che ci sono sia agli ingressi che alle uscite delle due porte NAND a 2 ingressi. In altre parole vogliamo che mostriate perché questo circuito è equivalente a una porta OR-esclusivo.

A
O
B
O
Q
A
O
B
1
Q
A
1
B
O
Q
A
1
B
1
Q

4. Disegnate i seguenti schemi:

   a. Porta AND-OR-INVERTITORE a 2 wide e 3 input

   b. Porta AND-OR-INVERTITORE a 4 wide e 2 input

5. Completate la seguente tabella della verità relativa al circuito sulla destra,

B	A	Q
0	0	
0	1	
1	0	
1	1	

e la seguente tabella della verità relativa al circuito sulla destra,

B	A	Q
1	0	
1	1	
0	0	
0	1	

Queste due rappresentazioni sono equivalenti rispetto alla funzione logica?

RISPOSTE

1. a.

Ingressi		Uscita	
B	A	Q	
0	0	1	
0	1	1	
1	0	1	*NAND*
1	1	0	

b.

Ingressi		Uscita	
B	A	Q	
0	0	0	
0	1	0	
1	0	0	*AND*
1	1	1	

c.

Ingressi		Uscita	
B	A	Q	
0	0	0	
0	1	0	
1	0	0	*AND*
1	1	1	

d.

Ingressi		Uscita	
B	A	Q	
0	0	0	
0	1	1	
1	0	1	*OR*
1	1	1	

e.

Ingressi		Uscita	
B	A	Q	
0	0	1	
0	1	1	
1	0	1	*NAND*
1	1	0	

f.

Ingressi		Uscita	
B	A	Q	
0	0	1	
0	1	0	
1	0	0	*NOR*
1	1	0	

g.

Ingressi		Uscita	
B	A	Q	
0	0	0	
0	1	0	
1	0	0	*AND*
1	1	1	

h.

Ingressi		Uscita	
B	A	Q	
0	0	1	
0	1	0	
1	0	0	*NOR*
1	1	0	

i.

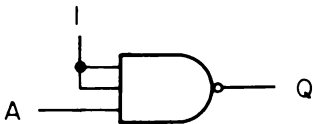
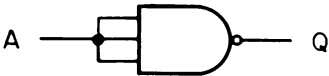
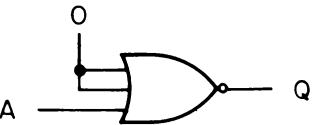
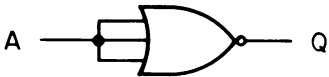
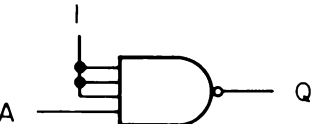
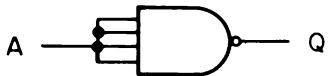
Ingressi		Uscita	
B	A	Q	
0	0	1	
0	1	1	
1	0	1	*NAND*
1	1	0	

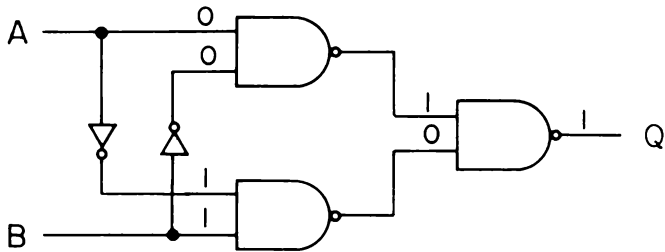
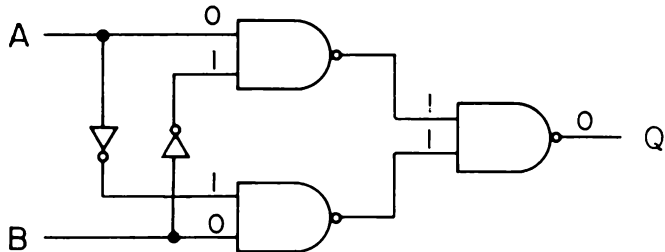
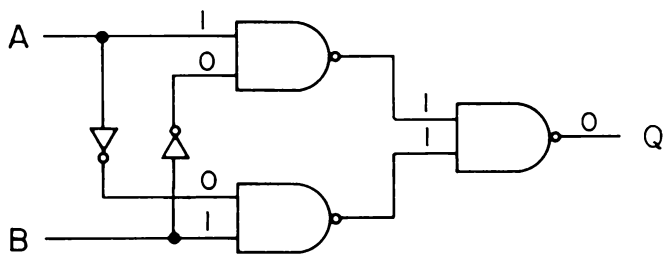
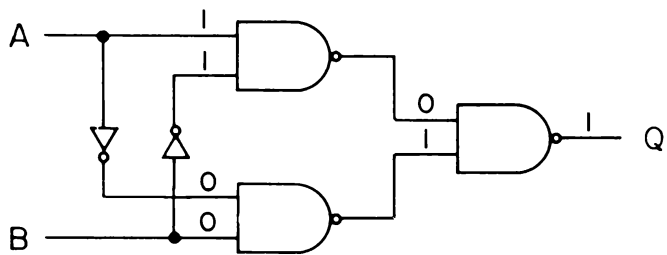
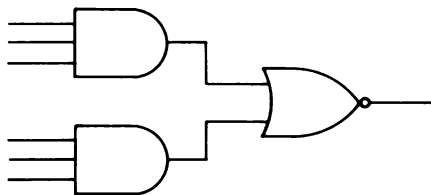
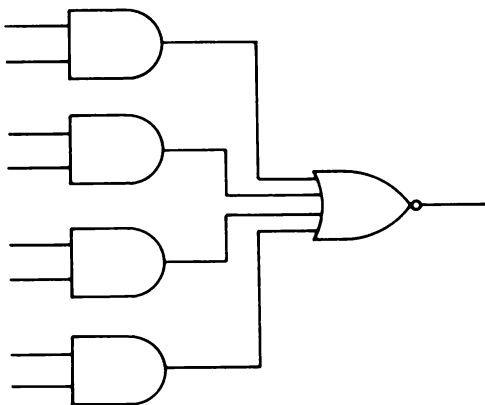
j.

Ingressi		Uscita	
B	A	Q	
0	0	0	
0	1	1	
1	0	1	*OR*
1	1	1	

k.

Ingressi		Uscita	
B	A	1	
0	0	1	
0	1	0	
1	0	0	*NOR*
1	1	0	

2. a.

b.

c.

d. Non è possibile convertire una porta AND a 4 ingressi in un invertitore.

3.

A
I
I
0
I
Q
0
0
B
A
I
0
I
I
0
Q
0
I
B
4. a.
b.

5. Primo circuito:

B	A	Q
0	0	1
0	1	0
1	0	0
1	1	0

Secondo circuito:

B	A	Q
1	0	0
1	1	1
0	0	0
0	1	0

No, non sono rappresentazioni equivalenti per la stessa funzione logica. Il primo è un NOR e il secondo è un AND.

CAPITOLO 8

ISTRUZIONI LOGICHE

INTRODUZIONE

Questo capitolo vi introdurrà alle istruzioni logiche, che sono preposte alla esecuzione delle operazioni logiche sulle parole binarie a 8 bit.

OBIETTIVI

Alla fine di questo capitolo sarete in grado di:

- Scrivere le tabelle della verità per le operazioni logiche a un bit AND, OR e XOR.
- Elencare gli esatti simboli Booleani per AND, OR, XOR e NOT.
- Spiegare come vengono eseguite le operazioni logiche su più bit.
- Eseguire le operazioni logiche AND, OR, XOR e NOT su una coppia di dati a 8 bit.
- Scrivere il teorema di De Morgan in algebra Booleana.
- Spiegare il teorema di De Morgan usando i simboli logici.
- Elencare le istruzioni logiche comprese nel set di istruzione dell'8080A.
- Spiegare come una istruzione logica può essere usata in un programma per microprocessor.
- Definire una maschera.

COSA È UNA ISTRUZIONE LOGICA?

Il microprocessor 8080 tratta informazioni a 8 bit in parallelo. Può spostare gli 8 bit da un registro interno ad un altro registro interno, da un registro interno alla memoria, oppure tra un accumulatore e un dispositivo di I/O esterno. Può anche eseguire operazioni aritmetiche e logiche, compreso *somme, complementi, sottrazioni,* tra le prime; *AND, OR* e *OR-Esclusivo,* tra le seconde.

In questo capitolo faremo la conoscenza con le tre operazioni logiche. *Un'operazione logica,* o *istruzione logica,* è eseguita su due dati da 8 bit, dove i bit corrispondenti di ogni dato sono sottoposti a operazioni logiche tipo AND, OR o OR-Esclusivo.
Nel microprocessor 8080, uno dei due dati è già presente nell'accumulatore; il risultato dell'operazione viene posto nell'accumulatore. È per questo che abbiamo appena detto che un accumulatore è chiamato "accumulatore"; infatti "accumula" i risultati delle operazioni logiche e aritmetiche che vengono via via eseguite.

Più avanti discuteremo come un microcomputer esegue le operazioni logiche AND, OR e OR-Esclusivo su dati a 8 bit; adesso esaminiamo come vengono eseguite le operazioni logiche base su un singolo bit prima di passare a quelle che coinvolgono 8 bit alla volta, in parallelo.

TABELLE DELLA VERITA' PER OPERAZIONI LOGICHE A UN BIT

Le regole base - tabelle della verità - che governano le operazioni logiche a un bit, sono già state trattate nel capitolo 7 ma le ripetiamo qui sotto:

AND

B	A	Q
0	0	0
0	1	0
1	0	0
1	1	1

OR

B	A	Q
0	0	0
0	1	1
1	0	1
1	1	1

XOR

B	A	Q
0	0	0
0	1	1
1	0	1
1	1	0

Queste tabelle sono dette "tabelle a un bit" perché le parole dei dati, A e B, contengono ciascuno un solo bit. XOR è una abbreviazione per OR-Esclusivo.

ALGEBRA BOOLEANA

Quando si parla di istruzioni logiche, è normale usare i *simboli Booleani,* i quali traggono origine dall'*Algebra Booleana,* che è la matematica dei sistemi logici. I simboli dell'alfabeto come A,B,C,.........Q sono usati per rappresentare le variabili logiche e i simboli 1 e 0 (stati logici). Questa matematica ebbe origine in Inghilterra nel 1847, per opera di George Boole, ma non entrò nell'uso comune fino al 1938, quando Claude Shannon l'impiegò per analizzare i circuiti a relé delle reti telefoniche.

Voi dovrete imparare solo i simboli base che vengono impiegati nei calcoli e tutti quelli della logica digitale. Tra questi simboli ci sono i seguenti:

+ indica *addizione logica (OR)*

indica *moltiplicazione logica (AND)*

+ indica *OR-Esclusivo (XOR)*

− indica *negazione (NOT)*

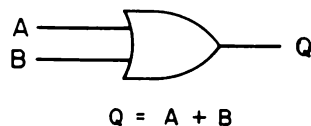
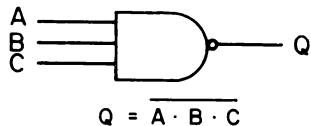
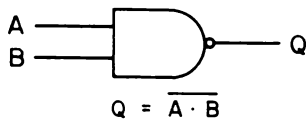
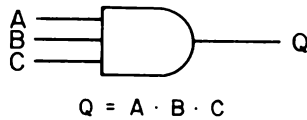
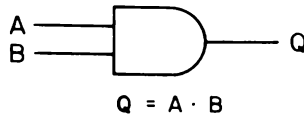
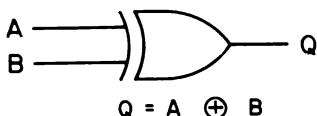
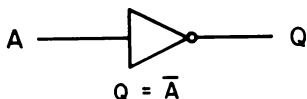
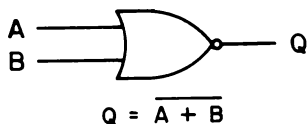
Il simbolo di negazione è un trattino sopra la variabile logica quale A,B,C,......Q. Allora, la notazione Booleana per una porta AND a 2 ingressi è Q=A·B o, più semplicemente, Q=AB, dove il simbolo di uguaglianza "=" sta a significare che le variabili, o gruppo di variabili, situate ai lati opposti del simbolo "=" sono equivalenti, cioè hanno lo stesso stato logico. Qui di seguito sono mostrate le notazioni Booleane per i diversi tipi di porte. Fate attenzione all'uso del trattino "−" per le porte NAND e NOR.

$Q = A \cdot B$

$Q = A \cdot B \cdot C$

$Q = \overline{A \cdot B}$

$Q = \overline{A \cdot B \cdot C}$

$Q = A + B$

Di solito si usa riassumere le notazioni simboliche, per le tre porte che abbiamo visto, nel modo seguente i

AND	OR	XOR
$0 \cdot 0 = 0$	$0 + 0 = 0$	$0 \oplus 0 = 0$
$0 \cdot 1 = 0$	$0 + 1 = 1$	$0 \oplus 1 = 1$
$1 \cdot 0 = 0$	$1 + 0 = 1$	$1 \oplus 0 = 1$
$1 \cdot 1 = 1$	$1 + 1 = 1$	$1 \oplus 1 = 0$

Queste sono le operazioni logiche a 1 bit.

OPERAZIONI LOGICHE A PIU' BIT

Le operazioni logiche a più bit sono eseguite nello stesso modo delle operazioni a un bit, e quindi non viene utilizzato alcun nuovo principio di logica. Un bit di una parola binaria opera logicamente sul corrispondente bit della seconda parola binaria producendo un risultato finale a più bit. La lunghezza della parola binaria può essere qualsiasi; 2 bit, 4 bit, 8 bit, 32 bit etc. Siccome il microprocessor 8080 esegue operazioni a più bit su parole di 8 bit, gli esempi saranno fatti con parole di questo tipo.

Considerate una variabile logica A a 8 bit; ogni singolo bit degli 8 che compongono la parola può essere indicato come: A7, A6, A5, A4, A3, A2, A1, e A0, dove A0 è il bit meno significativo (il bit corrispondente al peso 2^0) e dove il bit A7 è quello più significativo (quello corrispondente al peso 2^7). Inoltre considerate la variabile lógica B a 8 bit, composta dai bit B7, B6, B5, B4, B3, B2, B1 e B0 con B0 bit meno significativo e B7 bit più significativo.

Allora l'operazione logica, A·B=Q, significa:

$$A0 \cdot B0 = Q0$$

$$A1 \cdot B1 = Q1$$

$$A2 \cdot B2 = Q2$$

$$A3 \cdot B3 = Q3$$

$$A4 \cdot B4 = Q4$$

$$A5 \cdot B5 = Q5$$

$$A6 \cdot B6 = Q6$$

$$A7 \cdot B7 = Q7$$

Il risultato dell'operazione logica è la variabile logica Q che ha Q0 come bit meno significativo e Q7 come bit più significativo. In altre parole, *le operazioni logiche su più bit vengono eseguite bit per bit come una serie di operazioni logiche a un bit.*

Risulta più semplice eseguire le operazioni logiche se le parole binarie a più bit sono messe una sopra l'altra; quindi, se $A=11111111_2$ e $B=00010000_2$ allora A·B

```
11111111
00010000
--------
00010000
```

cioè $Q=00010000_2$. Abbiamo eseguito l'operazione logica AND e, per ottenere il risultato finale, abbiamo usato la relazione 0·1=0 e 1·1=1

In maniera del tutto analoga, l'operazione logica a più bit, A+B=Q ha il seguente significato:

$$A0 + B0 = Q0$$

$$A1 + B1 = Q1$$

$$A2 + B2 = Q2$$

$$A3 + B3 = Q3$$

$$A4 + B4 = Q4$$

$$A5 + B5 = Q5$$

$$A6 + B6 = Q6$$

$$A7 + B7 = Q7$$

Ancora una volta il risultato della operazione logica è la variabile logica Q, a 8 bit tale che, se $A=11111111_2$ e $B=00010000_2$, allora Q=A+B diventa,

```
11111111
00010000
--------
11111111
```

cioè $Q=11111111_2$. Abbiamo eseguito l'operazione logica OR e, per ottenere il risultato finale, abbiamo usato le relazioni 0+1=1 e 1+1=1. *Notate che il segno "+" rappresenta l'operazione logica OR e anche il "più" delle operazioni aritmetiche.*

L'ultima operazione logica che ci interessa, $A \oplus B = Q$, ha il seguente significato:

$$A0 + B0 = Q0$$

$$A1 + B1 = Q1$$

$$A2 + B2 = Q2$$

$$A3 + B3 = Q3$$

$$A4 + B4 = Q4$$

$$A5 + B5 = Q5$$

$$A6 + B6 = Q6$$

$$A7 + B7 = Q7$$

Il risultato di questa operazione di OR-Esclusivo è una variabile logica Q a 8 bit, tale che se $A=11111111_2$ e $B=00010000_2$, allora $Q = A \oplus B$ diventa,

$$\begin{array}{r} 11111111 \\ \underline{00010000} \\ 11101111 \end{array}$$

cioè $Q=11101111_2$. Abbiamo eseguito un OR-Esclusivo e, per ottenere il risultato finale, abbiamo usato le relazioni 0 + 1 = 1 e 1 + 1 = 0.

NOT

L'operazione logica NOT attiva il complemento di un qualsiasi digit o gruppo di digit binari. Se $A=11111111_2$, allora $Q=\bar{A}=00000000_2$; se $B=10011100_2$, allora $Q=\bar{B}=01100011_2$. Le due relazioni usate sono le seguenti

NOT

$$\bar{0} = 1$$

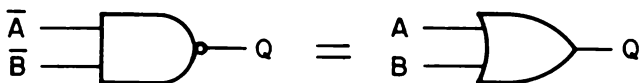
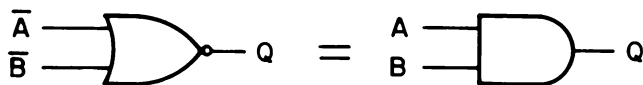
$$\bar{1} = 0$$

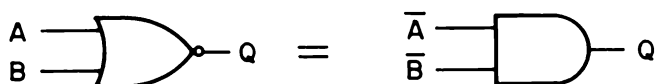
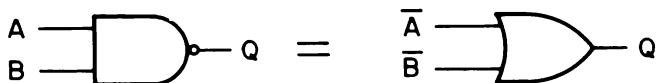
TEOREMA DI DE MORGAN

Un importante teorema dell'algebra Booleana è il *teorema di De Morgan*, che può essere rappresentato in due modi:

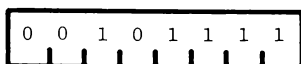
$$\overline{A \cdot B} = \bar{A} + \bar{B}$$

$$\overline{A + B} = \bar{A} \cdot \bar{B}$$

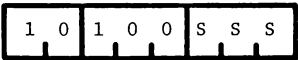
Un modo più interessante di enunciare il teorema di De Morgan è quello che fa uso dei simboli logici, cioè:

È un risultato molto importante; sarà molto usato nell'elettronica digitale e nell'interfacciamento dei micro-computer. Questo risultato vi permette di ottenere una funzione NOR negando tutti gli ingressi di una porta AND; e, d'altro canto, ottenere una funzione NAND negando tutti gli ingressi di una porta OR.

Il teorema di De Morgan può essere rappresentato anche dai seguenti simboli Booleani:

Questi simboli dimostrano che potete ottenere una funzione AND negando tutti gli ingressi di una porta NOR; e, d'altro canto, ottenere una funzione OR negando tutti gli ingressi di una porta NAND. I circuiti integrati contenenti una porta NAND sono molto comuni e poco costosi. Il teorema di De Morgan vi mostra come poter creare porte OR e NOR con le porte NAND.

COMPLEMENTA L'ACCUMULATORE: CMA

Complementare un accumulatore significa sostituire un bit allo stato logico 1 con un bit allo stato logico 0 e viceversa, cioè eseguire una operazione NOT su un accumulatore a 8 bit. L'istruzione, di un solo byte, che ha il codice ottale 057,

ha il codice mnemonico CMA ed un tempo di esecuzione di quattro cicli, o 5,333 μs per un microcomputer con frequenza 750 kHz.

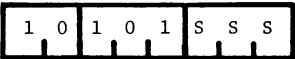
AND DI UN REGISTRO CON L'ACCUMULATORE: ANA

Le otto diverse istruzioni ANA comprese nel set dell'8080A hanno come codice mnemonico la forma generale ANA *S*, dove *S* è il registro sorgente, cioè il registro che contiene gli otto bit che vanno messi in AND con gli otto bit dell'accumulatore. L'istruzione è formata da un solo byte,

dove il valore SSS è dato dai tre bit che corrispondono al digit ottale che rappresenta il registro,

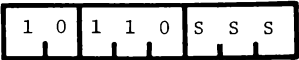
Registro	Codice ottale	SSS Codice binario
B	0	000
C	1	001
D	2	010
E	3	011
H	4	100
L	5	101
M	6	110
A	7	111

L'istruzione ANA M richiede una ulteriore spiegazione e sarà trattata nel capitolo successivo. Tutte le operazioni logiche richiedono per l'esecuzione quattro cicli, il che corrisponde a 5,333 μs. Le uniche eccezioni sono ANA M, ORA M e XRA M che richiedono sette cicli o 9,333 μs.

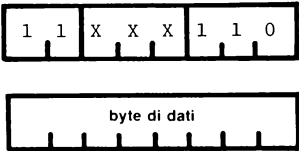
OR-ESCLUSIVO DI UN REGISTRO CON L'ACCUMULATORE: XRA

Il codice mnemonico generico, XRA *S,* è valido per le otto istruzioni dell'8080A che eseguono l'operazione logica di OR-Esclusivo tra il contenuto di un registro e il contenuto dell'accumulatore. Il risultato di questa operazione logica, come il risultato delle istruzioni ANA e ORA è messo nell'accumulatore, cioè il contenuto precedente dell'accumulatore è sostituito dal risultato dell'operazione logica. Il contenuto del registro *non cambia* ed è per questo motivo che l'accumulatore nei microcomputer è considerato un registro speciale. L'istruzione XRA che è di un solo byte,

ha un tempo di esecuzione di quattro cicli, o 5,333 μs, con l'eccezione, come abbiamo già detto, della istruzione XRA M, che viene eseguita in 7 cicli.

OR DI UN REGISTRO CON L'ACCUMULATORE: ORA

La terza ed ultima istruzione logica è la ORA. Le otto istruzioni ORA hanno come codice mnemonico generale ORA *S*, dove *S* è il registro che contiene gli otto bit che devono essere messi in OR con gli otto bit contenuti nell'accumulatore. Il risultato dell'operazione logica di OR è messo nell'accumulatore mentre il contenuto del registro rimane invariato. Come la ANA e la XRA, anche questa istruzione è di un solo byte,

ed ha un tempo di esecuzione di quattro cicli, o 5,333 μs, con l'eccezione della ORA M, che viene eseguita in 7 cicli.

OPERAZIONI LOGICHE IMMEDIATE: ANI, XRI, ORI

Le tre operazioni logiche immediate sono istruzioni di due byte,

dove XXX sono i bit che corrispondono allo specifico digit di codice ottale relativo al tipo di operazione logica

Mnemonico	Operazione logica	Codice ottale	XXX Codice binario
ANI	AND	4	100
XRI	XOR	5	101
ORI	OR	6	110

Allora abbiamo:

AND immediato con l'accumulatore: 346 <B2>

OR-Esclusivo immediato con l'accumulatore: 356 <B2>

OR immediato con l'accumulatore: 366 <B2>

Il simbolo <B2> rappresenta il secondo byte dell'istruzione, cioè il byte del dato.

Mediante l'*operazione logica immediata* si fa in modo che il dato contenuto nel secondo byte dell'istruzione venga ad operare logicamente sul contenuto dell'accumulatore, e il risultato si trova nell'accumulatore stesso. Le operazioni possono essere AND, OR o OR-Esclusivo. Ognuna di esse richiede sette cicli, o 9,333 μs.

RIASSUNTO DELLE ISTRUZIONI LOGICHE

La tabella dei codici operativi che abbiamo già mostrato nel capitolo 6, viene ripetuta nella pagina seguente. Certamente riuscirete a trovare le istruzioni ANA, ORA, XRA, ANI, ORI e XRI e i corrispondenti codici operazione in ottale. Le strutture base del codice ottale di questi tipi di istruzioni sono le seguenti:

Codice operazione per istruzione logica su un registro: 2 X S

Codice operazione per istruzione logica immediata: 3 X 6 <B2>

dove *S* è il codice ottale di un digit corrispondente al registro sorgente e *X* è il codice ottale di un digit corrispondente all'operazione logica, cioè *X* è uguale a 4 per AND, 5 per OR-Esclusivo e 6 per OR.

PERCHÉ ABBIAMO BISOGNO DELLE ISTRUZIONI LOGICHE?

Adesso che sapete cos'è una operazione logica, potete domandare: perchè abbiamo bisogno di inserire istruzioni logiche nei programmi dei microcomputer? Una delle risposte è che *le istruzioni logiche permettono di stabilire quali dispositivi esterni sono in on o in off oppure quale specifico evento si è verificato oppure no.* Per esempio, supponiamo che si usi lo 0 logico e l'1 logico per rappresentare uno dei seguenti tipi di operazione:

A. Stato on/off di un dispositivo

Stato logico 0 = dispositivo in off
Stato logico 1 = dispositivo in on

B. Verificarsi di un evento

Stato logico 0 = l'evento non si è verificato
Stato logico 1 = l'evento si è verificato

Nel capitolo seguente imparerete come usare l'istruzione IN per introdurre otto bit di dati nell'accumulatore, e imparerete anche che potete usare ogni singolo bit per rappresentare lo stato on/off di un dispositivo, oppure il verificarsi o il non verificarsi di un evento. Con otto bit potete quindi rappresentare otto diversi dispositivi o eventi. Considerate i seguenti otto dispositivi che possono essere in off o in on:

BIT 0: dispositivo misuratore di pressione
BIT 1: dispositivo misuratore di temperatura
BIT 2: dispositivo misuratore di velocità

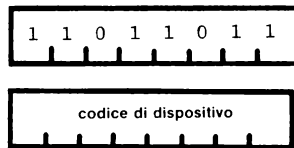
	0	1	2	3	4	5	6	7
00	NOP	LXI B **	STAX B	INX B	INR B	DCR B	MVI B *	RLC
01	—	DAD B	LDAX B	DCX B	INR C	DCR C	MVI C *	RRC
02	—	LXI D **	STX D	INX D	INR D	DCR D	MVI D *	RAL
03	—	DAD D	LDAX D	DCX D	INR E	DCR E	MVI E *	RAR
04	—	LXI H **	SHLD **	INX H	INR H	DCR H	MVI H *	DAA
05	—	DAD H	LHLD **	DCX H	INR L	DCR L	MVI L *	CMA
06	—	LXI SP **	STA **	INX SP	INR M	DCR M	MVI M *	STC
07	—	DAD SP	LDA **	DCX SP	INR A	DCR A	MVI A *	CMC
	0	1	2	3	4	5	6	7
10	MOV B,B	MOV B,C	MOV B,D	MOV B,E	MOV B,H	MOV B,L	MOV B,M	MOV B,A
11	MOV C,B	MOV C,C	MOV C,D	MOV C,E	MOV C,H	MOV C,L	MOV C,M	MOV C,A
12	MOV D,B	MOV D,C	MOV D,D	MOV D,E	MOV D,H	MOV D,L	MOV D,M	MOV D,A
13	MOV E,B	MOV E,C	MOV E,D	MOV E,E	MOV E,H	MOV E,L	MOV E,M	MOV E,A
14	MOV H,B	MOV H,C	MOV H,D	MOV H,E	MOV H,H	MOV H,L	MOV H,M	MOV H,A
15	MOV L,B	MOV L,C	MOV L,D	MOV L,E	MOV L,H	MOV L,L	MOV L,M	MOV L,A
16	MOV M,B	MOV M,C	MOV M,D	MOV M,E	MOV M,H	MOV M,L	HLT	MOV M,A
17	MOV A,B	MOV A,C	MOV A,D	MOV A,E	MOV A,H	MOV A,L	MOV A,M	MOV A,A
	0	1	2	3	4	5	6	7
20	ADD B	ADD C	ADD D	ADD E	ADD H	ADD L	ADD M	ADD A
21	ADC B	ADC C	ADC D	ADC E	ADC H	ADC L	ADC M	ADC A
22	SUB B	SUB C	SUB D	SUB E	SUB H	SUB L	SUB M	SUB A
23	SBB B	SBB C	SBB D	SBB E	SBB H	SBB L	SBB M	SBB A
24	ANA B	ANA C	ANA D	ANA E	ANA H	ANA L	ANA M	ANA A
25	XRA B	XRA C	XRA D	XRA E	XRA H	XRA L	XRA M	XRA A
26	ORA B	ORA C	ORA D	ORA E	ORA H	ORA L	ORA M	ORA A
27	CMP B	CMP C	CMP D	CMP E	CMP H	CMP L	CMP M	CMP A
	0	1	2	3	4	5	6	7
30	RNZ	POP B	JNZ **	JMP **	CNZ **	PUSH B	ADI *	RST 0
31	RZ	RET	JZ **	—	CZ **	CALL **	ACI *	RST 1
32	RNC	POP D	JNC **	OUT *	CNC **	PUSH D	SUI *	RST 2
33	RC	—	JC **	IN *	CC **	—	SBI *	RST 3
34	RPO	POP H	JPO **	XTHL	CPO **	PUSH H	ANI *	RST 4
35	RPE	PCHL	JPE **	XCHG	CPE **	—	XRI *	RST 5
36	RP	POP PSW	JP **	DI	CP **	PUSH PSW	ORI *	RST 6
37	RM	SPHL	JM **	EI	CM **	—	CPI *	RST 7

Tabella dei codici di operazione sviluppata da R. Baker [*Byte 84* (Febbraio 1976)]   Ogni * indica un byte di istruzione in più.

BIT 3: dispositivo misuratore di flusso
BIT 4: dispositivo misuratore di tensione
BIT 5: dispositivo misuratore di corrente
BIT 6: dispositivo misuratore del numero di mucche
BIT 7: dispositivo misuratore di angeli-sulla-punta-di-uno-spillo.

Questi otto dispositivi sono associati a otto bit che vengono posti in ingresso, tutti insieme, allo stesso momento, nell'accumulatore del microprocessor 8080A e, una volta che sono memorizzati, potete usare le istruzioni logiche per determinare quali siano in on o in off. Dopo avervi spiegato un'altra istruzione del microcomputer, l'istruzione IN, vi proporremo alcuni programmi che verificano gli stati on/off dei dispositivi che abbiamo visto.

CARICA NELL'ACCUMULATORE DEI DATI DA UN DISPOSITIVO DI INGRESSO: IN

Come l'istruzione OUT, questa è una delle istruzioni più importanti del set del microprocessor 8080A. Infatti vi permette di mettere in ingresso otto bit di dati da un dispositivo esterno nell'accumulatore. Il dispositivo esterno viene selezionato mediante il contenuto del secondo byte dell'istruzione, il *codice di selezione dispositivo.* Un altro aspetto importante di questa istruzione è che permette di generare un singolo impulso di sincronizzazione, detto *impulso di selezione dispositivo,* il quale sincronizza lo scambio dei dati tra un dispositivo esterno e l'accumulatore. Il codice mnemonico di questa istruzione è IN "n" e il suo codice operazione è 333,

essa richiede, per l'esecuzione, 10 cicli, cioè 13,333 μs per un microcomputer che opera a 750 kHz.

Nel contesto dell'uso delle istruzioni logiche, l'istruzione IN è determinante perché vi permette di mettere in input otto bit distinti, sia da un singolo dispositivo, sia da otto diversi dispositivi ciascuno dei quali è associato ad almeno un bit: Il bit 0 per il dispositivo misuratore di pressione, il bit 1 per il dispositivo misuratore di temperatura, il bit 2 per il dispositivo misuratore di velocità etc.

PRIMO PROGRAMMA

In questo programma vi porrete la seguente domanda: Il dispositivo misuratore di flusso è in on? Se la risposta è si, l'uscita sulla porta 0 dovrà essere 010; se è no, l'uscita sulla porta 0 dovrà essere 000.

Indirizzo di memoria LO	Byte di istruzioni	Codice mnemonico	Descrizione.
207	333	IN	Istruzione di due byte che carica gli otto bit di on/off nell'accumulatore.
210	003	003	Con un microcomputer completamente interfacciato, questa è una istruzione IN con codice dispositivo associato, come viene indicato in questo caso. I due byte di istruzione mettono in ingresso un byte di otto bit dal dispositivo di ingresso numero 3 nell'accumulatore.
211	346	ANI	Metti in modo immediato in AND il byte successivo con contenuto dell'accumulatore.
211	346	ANI	Metti in modo immediato in AND il byte successivo con contenuto dell'accumulatore.
212	010	010	Byte di mascheratura che "maschera" tutti i bit tranne il bit 3.
213	323	OUT	Metti in uscita il contenuto dell'accumulatore sulla porta 0
214	000	000	Codice di dispositivo della porta 0
215	166	HLT	Alt del microcomputer.

Il termine, *maschera* viene definito come segue:

Maschera tecnica mediante la quale certi bit di una parola di più bit vengono annullati o inibiti.

Una maschera per il viso copre alcune parti della faccia, allo stesso modo, una maschera, usata in una operazione da un computer, copre alcuni o quasi tutti i bit di una parola di più bit, lasciando scoperti solo quei bit che sono importanti alla prosecuzione del programma.

Ritornando all'esempio che abbiamo fatto e supponendo che tutti i dispositivi sono in on, allora gli otto bit del byte che indica lo stato dei dispositivi, se sono tutti ad 1, danno il valore 377. In questo caso, agli indirizzi LO 211 e 213, si avrebbe la seguente operazione logica:

11111111	Contenuto dell'accumulatore
00001000	Byte in AND immediato con l'accumulatore
00001000	Risultato dell'operazione logica AND

e il risultato logico appare sulla porta 0; questo risultato ci indica che il dispositivo misuratore di flusso era in on al momento in cui il byte indicatore di stato è stato caricato nell'accumulatore.

A questo punto non vi abbiamo ancora dimostrato come mettere in ingresso otto bit di dato sul microcomputer MMD-1 e quindi non siete ancora in grado di eseguire questo programma. Continuate col prossimo programma, il secondo, dove escluderemo questo problema rimpiazzando l'istruzione IN 3 con una MVI A; questa sostituzione ci permette di simulare il caricamento di un byte da dei dispositivi esterni nell'accumulatore.

VARIAZIONE AL PRIMO PROGRAMMA

Questa variazione ci permetterà di eseguire sul microcomputer MMD-1, il programma che abbiamo illustrato. Una istruzione MVI A va a sostituire la IN 3 agli indirizzi di memoria LO=207 e 210. Ricordiamo che per questi esempi HI è sempre uguale a 003.

Indirizzo di memoria LO	Byte di istruzione	Codice mnemonico	Descrizione
207	067	MVI A	Metti in modo immediato il byte seguente nell'accumulatore
210	<B2>	<B2>	Simulazione del byte di stato del dispositivo; può assumere qualsiasi valore tra 000 e 377
211	346	ANI	Metti in modo immediato in AND il byte successivo col contenuto dell'accumulatore
212	010	010	Byte di maschera che "maschera" tutti i bit eccetto il bit 3
213	323	OUT	Metti in uscita il contenuto dell'accumulatore sulla porta 0
214	000	000	Codice di dispositivo della porta 0
215	166	HLT	Alt del microcomputer

Il byte di istruzione <B2> è il byte di tre digit ottali di stato dei dispositivi, che indica quali degli otto dispositivi sono in on e quali sono in off. Se il contenuto del byte fosse 147, il dispositivo misuratore di flusso sarebbe in on? La risposta è no perché nel byte di stato al valore 147 il bit 3 è 0, quindi l'esecuzione del programma con il valore 147 all'indirizzo LO=210, darebbe un'uscita sulla porta 0 pari a 000.

Usando maschere diverse si può verificare se un qualsiasi dispositivo è in on e si può verificarne uno per volta o due o comunque un numero qualsiasi fino a otto alla volta. Si può quindi concludere che:

Se il bit *n* sulla porta 0 è 0: Il dispositivo era in off

Se il bit *n* sulla porta 0 è 1: Il dispositivo era in on

SECONDO PROGRAMMA

Adesso la domanda è: Il dispositivo misuratore di flusso è in off?
Il programma che risponde a questa domanda è il seguente:

Indirizzo di memoria LO	Byte di istruzione	Codice mnemonico	Descrizione
216	076	MVI A	Metti in modo immediato il byte successivo nell'accumulatore
217	<B2>	<B2>	Byte che simula lo stato dei dispositivi
220	057	CMA	Complementa il contenuto dell'accumulatore
221	346	ANI	Metti in modo immediato in AND il byte seguente con contenuto dell'accumulatore
222	010	010	Maschera tutti i bit eccetto il 3
223	323	OUT	Metti in uscita il contenuto dell'accumulatore sulla porta 0
224	000	000	Codice di dispositivo della porta 0
225	166	HLT	Alt del microcomputer

Col byte di maschera all'indirizzo LO=222 è possibile verificare se uno qualsiasi dei dispositivi era in on in corrispondenza dell'ingresso del byte di stato nell'accumulatore ed è possibile verificare un solo dispositivo o più dispositivi contemporaneamente. Questa volta il risultato è che:

Se il bit *n* sulla porta 0 è 0: Il dispositivo era in on

Se il bit *n* sulla porta 0 è 1: Il dispositivo era in off

TERZO PROGRAMMA

La terza domanda che ci si può porre è: Il dispositivo misuratore di flusso ha cambiato stato dall'ultima volta che è stato controllato? Nel programma che segue, l'ultimo stato dei dispositivi è stato memorizzato nel registro B e lo stato attuale è messo nell'accumulatore.

Indirizzo di memoria LO	Byte di istruzioni	Codice mnemonico	Descrizione
226	006	MVI B	Metti in modo immediato il byte successivo nel registro B
227	<B2>	<B2>	Byte di stato precedente

230	076	MVI A	Metti in modo immediato il byte successivo nell'accumulatore
231	<B2>	<B2>	Byte di stato attuale
232	250	XRA B	Metti in OR-Esclusivo il contenuto del registro B col contenuto dell'accumulatore
233	323	OUT	Metti in uscita il contenuto dell'accumulatore sulla porta 0
234	000	000	Codice di dispositivo della porta 0
235	166	HLT	Alt del microcomputer

Con questo programma è possibile vedere quali dispositivi hanno cambiato stato tra l'ultima verifica e l'attuale. Ricordate che i due byte di stato sono stati prelevati in due momenti diversi e il microcomputer controlla lo stato dei dispositivi solo quando memorizza il byte tramite una istruzione IN, che nel nostro programma viene simulata dalla MVI A mentre avevamo supposto di avere il byte di stato precedente nel registro B. Allora le conclusioni sono:

Se il bit *n* della porta 0 è 0: Il dispositivo non ha cambiato stato

Se il bit *n* della porta 0 è 1: Il dispositivo ha cambiato stato

QUARTO PROGRAMMA

Adesso vogliamo determinare se un certo dispositivo ha cambiato stato o no e sapere se è passato in on o in off. Un programma che risponde a queste domande è il seguente:

Indirizzo di memoria LO	Byte di istruzioni	Codice mnemonico	Descrizione
236	006	MVI B	Metti in modo immediato il byte seguente nel registro B
237	<B2>	<B2>	Byte di stato precedente
240	076	MVI A	Metti in modo immediato il byte seguente nell'accumulatore
241	<B2>	<B2>	Byte di stato attuale
242	117	MOV C,A	Trasferisci il contenuto dell'accumulatore nel registro C. Il contenuto dell'accumulatore resta inalterato
243	250	XRA B	Metti in OR-Esclusivo il contenuto del registro B con l'accumulatore
244	127	MOV D,A	Trasferisci il nuovo contenuto dell'accumulatore nel registro D. Il contenuto dell'accumulatore resta immutato
245	240	ANA B	Metti in AND il contenuto del registro B col contenuto dell'accumulatore

246	323	OUT	Metti in uscita il contenuto dell'accumulatore sulla porta 0
247	000	000	Codice di dispositivo della porta 0
250	057	CMA	Complementa il contenuto dell'accumulatore
251	242	ANA D	Metti in AND il contenuto del registro D con l'accumulatore
252	323	OUT	Metti in uscita il contenuto dell'accumulatore sulla porta 1
253	001	001	Codice di dispositivo della porta 1
254	166	HLT	Alt del microcomputer

I risultati che questo programma produce possono essere così riassunti:

Se il bit *n* sulla porta 0 è 0: Il dispositivo non è passato da on a off.

Se il bit *n* sulla porta 0 è 1: Il dispositivo è passato da on a off

Se il bit *n* sulla porta 1 è 0: Il dispositivo non è passato da off a on

Se il bit *n* sulla porta 1 è 1: Il dispositivo è passato da off a on.

Allora, ricapitolando, è possibile per mezzo delle istruzioni logiche, scrivere dei programmi per il microcomputer che rispondano ai seguenti tipi di domande:

- Il bit di stato è 0 o 1?
- Il bit di stato è cambiato rispetto ad un valore precedente oppure no?
- Se il bit di stato è cambiato, lo ha fatto passando da 0 a 1 o da 1 a 0?

INTRODUZIONE AGLI ESPERIMENTI

I seguenti esperimenti sono l'esecuzione di alcuni dei semplici programmi che sono stati descritti in questo capitolo e vi permettono di fare esperienza sull'uso delle istruzioni logiche per determinare lo stato ed i cambiamenti di stato dei bit che lo rappresentano.
In ogni programma, l'ingresso del byte di stato viene simulato con delle appropriate istruzioni MVI.

Gli esperimenti che farete possono essere così riassunti:

Esperimento N.	Commento
1	Eseguire un programma molto semplice che determini quale degli otto diversi dispositivi sia in on e che mostri l'uso delle maschere
2	Dimostrare l'uso delle istruzioni logiche per determinare se un dispositivo ha cambiato stato e, se lo ha cambiato, in quale direzione.

ESPERIMENTO N. 1

Scopo

Lo scopo di questo esperimento è di determinare quali dei seguenti otto dispositivi,

Bit 0: dispositivo misuratore di pressione
Bit 1: dispositivo misuratore di temperatura
Bit 2: dispositivo misuratore di velocità
Bit 3: dispositivo misuratore di flusso
Bit 4: dispositivo misuratore di tensione
Bit 5: dispositivo misuratore di corrente
Bit 6: dispositivo misuratore di livello di liquidi
Bit 7: dispositivo misuratore di frequenza

sono in on. Lo stato logico 1 indica che il dispositivo è in on, e lo stato logico 0 indica che il dispositivo è in off. L'ingresso del byte di stato viene simulato con una istruzione MVI A.

Programma

Indirizzo di memoria LO	Byte di istruzione	Codice mnemonico	Descrizione
207	076	MVI A	Metti in modo immediato il byte seguente nell'accumulatore
210	245	245	Byte di stato per otto dispositivi
211	346	ANI	Metti in modo immediato in AND il byte seguente col contenuto dell'accumulatore
212	377	377	Byte di maschera che non esclude nessuno degli otto dispositivi
213	323	OUT	Metti in uscita il contenuto dell'accumulatore sulla porta 1
214	001	001	Codice di dispositivo della porta 1
215	166	HLT	Alt del microcomputer

Passo 1

Caricate ed eseguite sul microcomputer questo programma. Riteniamo che ormai avete abbastanza esperienza con i tasti RESET, H,L,G e S da non aver più bisogno di istruzioni che vi dicano come e quando usarli.

Guardate gli indicatori a LED della porta 1. Quali dei dispositivi che abbiamo simulato è in on?

Sono in on i dispositivi misuratori di frequenza, corrente, velocità, pressione e tutti gli altri sono in off. Corrisponde?

Passo 2

Sostituite il valore del byte di stato a LO = 210 con 133 ed eseguite il programma una seconda volta. Quali dispositivi sono in on adesso?

Sono in on i dispositivi misuratori di livello di liquidi, tensione, flusso, temperatura e pressione; tutti gli altri sono in off.

Passo 3

Sostituite il valore del byte di stato a LO=210 con 070 ed eseguite il programma una terza volta. Quali dispositivi sono in on adesso?

Sono in on i dispositivi misuratori di flusso, tensione e corrente; tutti gli altri sono in off.

Passo 4

Sostituite il valore del byte di maschera a LO=212 con 020. Quali bit di stato state eliminando (mascherando)?

Tutti i bit di stato tranne il bit 4 che corrisponde al dispositivo misuratore di tensione.

Eseguite il programma e dite cosa vedete in uscita sulla porta 001.

Il bit di stato 4 è ad 1 ed indica quindi che il dispositivo misuratore di tensione è in on.

ESPERIMENTO N. 2

Scopo

Lo scopo di questo esperimento è determinare quali degli otto dispositivi che abbiamo elencato nell'esperimento n. 1 sono in on, quali hanno cambiato stato da on a off e quali hanno cambiato stato da off a on. Il contenuto del registro B simula il byte di stato precedente.

Programma

Indirizzo di memoria LO	Byte di istruzione	Codice mnemonico	Descrizione
236	006	MVI B	Metti in modo immediato il byte seguente nel registro B
237	210	210	Byte di stato precedente
240	076	MVI A	Metti in modo immediato il byte seguente nell'accumulatore
241	011	011	Byte di stato attuale
242	117	MOV C,A	Trasferisci il contenuto dell'accumulatore nel registro C.
243	150	XRA B	Metti in OR-Esclusivo il contenuto del registro B con l'accumulatore
244	127	MOV D,A	Trasferisci il contenuto dell'accumulatore nel registro D
245	240	ANA B	Metti in AND il contenuto dell'accumulatore col contenuto del registro B
246	323	OUT	Metti in uscita il contenuto dell'accumulatore sulla porta 0
247	000	000	Codice di dispositivo della porta 0
250	057	CMA	Complementa il contenuto dell'accumulatore
251	242	ANA D	Metti in AND il contenuto del registro D con l'accumulatore
252	323	OUT	Metti in uscita il contenuto dell'accumulatore sulla porta 1
253	001	001	Codice di dispositivo della porta 1
254	172	MOV A,D	Trasferisci il contenuto del registro C nell'accumulatore
255	323	OUT	Metti in uscita il contenuto dell'accumulatore sulla porta 2
256	002	002	Codice di dispositivo della porta 2
257	166	HLT	Alt del microcomputer

Passo 1

Prima di caricare ed eseguire il programma, fate le seguenti operazioni logiche:

(a) 10001000 XOR 00001001

```
10001000
00001001
--------
```

Il risultato di questa operazione logica vi dice quali dispositivi hanno cambiato stato.

(b) 10000001 AND 10001000

```
10001000
10000001
--------
```

Il risultato di questa operazione logica vi dice quali dispositivi hanno cambiato stato da on a off.

(c) NOT 10000000, il complemento di 10000000

(d) 01111111 AND 10000001

```
01111111
10000001
--------
```

Il risultato di questa operazione logica vi dice quali dispositivi hanno cambiato stato da off a on.

Passo 2

Caricate ed eseguite questo programma sul microcomputer. Quale informazione appare sugli indicatori a LED della porta 1?

Quei dispositivi che hanno cambiato stato da off a on.

Passo 3

Quale informazione appare in uscita sulla porta 0?

Quei dispositivi che hanno cambiato stato da on a off.

Passo 4

Quale informazione appare in uscita sulla porta 2?

L'attuale byte di stato, cioè, quei dispositivi che sono in on e quelli che sono in off.

DOMANDE RIEPILOGATIVE

Le seguenti domande vi aiuteranno a ripassare l'uso delle istruzioni logiche, l'algebra Booleana, e le operazioni logiche su più bit.

1. Eseguite le operazioni su più bit indicate

   a. 11001011 · 01011010

   b. 00100000 + 11011111

   c. 00100000 · 11011111

   d. 10101010 + 10100100

   e. $314_8 \cdot 010_8$

   f. $246_8 + 200_8$

   g. $077_8 + 004_8$

   h. $111_8 \cdot 033_8$

2. Un byte di stato di otto bit è associato ad otto diversi dispositivi:

   Bit 0: dispositivo misuratore di pressione
   Bit 1: dispositivo misuratore di temperatura
   Bit 2: dispositivo misuratore di velocità
   Bit 3: dispositivo misuratore di flusso
   Bit 4: dispositivo misuratore di tensione
   Bit 5: dispositivo misuratore di corrente
   Bit 6: dispositivo misuratore di livello di liquidi
   Bit 7: dispositivo misuratore di frequenza

Per ogni valore del byte di stato indicato qui sotto, dite quali di questi dipositivi sono in on e quali sono in off.
Ogni bit allo stato logico 1 indica che il dispositivo è in on.

   a. 123_8

   b. 100_8

   c. 144_8

   d. 040_8

   e. 002_8

   f. 060_8

   g. 006_8

   h. 300_8

   i. 001_8

   j. 050_8

3. Con ogni coppia di byte di stato indicata, byte di stato precedente e byte di stato attuale, usate l'algebra Booleana per determinare quali degli otto diversi dispositivi che abbiamo elencato nella seconda domanda hanno cambiato stato da on a off o da off a on. Scrivete anche i calcoli di algebra Booleana che fate.

Byte di stato precedente	Byte di stato attuale
a. 204	106
b. 057	143
c. 002	007
d. 247	333

RISPOSTE

1. a. 01001010
   b. 11111111
   c. 00000000
   d. 00001110
   e. 00001000 = 010_8
   f. 00100110 = 046_8
   g. 00111111 = 077_8
   h. 00001001 = 011_8

2. a. Dispositivi misuratori di livello di liquidi, tensione, temperatura e pressione
   b. Dispositivo misuratore di livello di liquidi
   c. Dispositivi misuratori di livello di liquidi, corrente e velocità.
   d. Dispositivo misuratore di corrente
   e. Dispositivo misuratore di temperatura
   f. Dispositivi misuratori di corrente e tensione
   g. Dispositivi misuratori di velocità e temperatura
   h. Dispositivi misuratori di frequenza e livello di liquidi
   i. Dispositivo misuratore di pressione
   j. Dispositivi misuratori di corrente e flusso.

3. a. Prima dovete convertire i due byte ottali in codice binario.

   204_8 = 10000100 (byte di stato precedente)
   106_8 = 01000110 (byte di stato attuale)

   Poi eseguite una operazione di OR-Esclusivo in questi due byte

   $10000100 + 01000110 = 11000010$

   Mettete adesso in AND il risultato, 11000010, con il byte di stato precedente.

   $10000100 \cdot 11000010 = 10000000$

   Eseguite poi una operazione NOT sul risultato,

   $\overline{10000000} = 01111111$

   Infine usate il risultato della operazione NOT e mettetelo in AND col risultato della operazione di OR-Esclusivo fatta inizialmente,

   $01111111 \cdot 11000010 = 01000010$

   Adesso possiamo tirare le conclusioni.

   1. I dispositivi misuratori di frequenza, livello di liquidi e temperatura hanno cambiato stato.

   2. Il dispositivo misuratore di frequenza è passato da on a off.

   3. I dispositivi misuratori di livello di liquidi e temperatura sono passati da off a on.

Guardando i due byte di stato, possiamo concludere dicendo che l'algebra Booleana ci ha fornito le risposte esatte.

b. Convertite i byte di stato in codice binario,

057_8 = 00101111 (byte di stato precedente)
143_8 = 01100011 (byte di stato attuale)

Eseguite una operazione di OR-Esclusivo,

$00101111 \oplus 01100011 = 01001100$

Usate il risultato per metterlo in AND col byte di stato precedente,

$00101111 \cdot 01001100 = 00001100$

Possiamo concludere che i dispositivi misuratori di livello di liquidi, flusso e velocità hanno cambiato stato; inoltre i dispositivi misuratori di flusso e velocità sono passati da on a off.

Complementate il risultato dell'operazione AND,

$\overline{00001100} = 11110011$

Mettete in AND questo risultato col risultato della operazione iniziale XOR,

$11110011 \cdot 01001100 = 01000000$

Si vede che il dispositivo misuratore di livello di liquidi è passato da off a on.

c. Convertite i byte di stato in codice binario,

002_8 = 00000010 (byte di stato precedente)
007_8 = 00000111 (byte di stato attuale)

Fate una operazione di XOR,

$00000010 \odot 00000111 = 00000101$

Usate il risultato per metterlo in AND col byte di stato precedente,

$00000010 \cdot 00000101 = 00000000$

Quindi, i dispositivi misuratori di pressione e velocità hanno cambiato stato e nessuno di essi è passato da on a off.

Complementate il risultato della operazione AND,

$\overline{00000000} = 11111111$

Mettete in AND questo risultato col risultato dell'operazione XOR iniziale,

$11111111 \cdot 00000101 = 00000101$

Se ne decuce, come era evidente, che sia il dispositivo misuratore di velocità che quello misuratore di pressione sono passati da off a on.

d. Convertite i byte di stato in codice binario,

247_8 = 10100111 (byte di stato precedente)
333_8 = 11011011 (byte di stato attuale)

Eseguite una operazione di XOR,

$10100111 \oplus 11011011 = 01111100$

Usate il risultato per metterlo in AND col byte di stato precedente,

$01111100 \cdot 10100111 = 00100100$

Ne deduciamo che hanno cambiato stato i dispositivi misuratori di livello di liquidi, corrente, tensione, flusso e velocità e che quelli misuratori di corrente e velocità sono passati da on a off.

Complementate il risultato dell'operazione AND,

$\overline{00100100} = 11011011$

Mettete in AND questo risultato e il risultato dell'operazione iniziare XOR,

$11011011 \cdot 01111100 = 01011000$

Allora i dispositivi misuratori di livello di liquidi, tensione e flusso sono passati da off a on.

CAPITOLO 9

INTRODUZIONE AL BREADBOARDING

INTRODUZIONE

Questo capitolo vi fornirà i concetti introduttivi sull'uso della *Solderless Breadboard*: una piastra di montaggio a matrice sulla quale vengono collocati circuiti integrati, componenti elettronici e relativi cavi di collegamento. Alcuni semplici esperimenti vi consentiranno di fare esperienza con circuiti montati su piastre separate denominati Outboards, e capaci di svolgere *funzioni ausiliarie.* Si tratta di dispositivi digitali di ingresso-uscita che vi permettono di introdurre in un circuito delle informazioni digitali e poi riconoscere e visualizzare le informazioni che ne escono. Vi parleremo anche dell'uso di simboli e schemi come rappresentazione simbolica dei circuiti digitali.

OBIETTIVI

Alla fine di questo capitolo sarete in grado di:

- Dare la definizione di breadbording.
- Spiegare le caratteristiche della solderless breadboard, (piastra senza saldature), SK-10.
- Dare la definizione di funzione ausiliaria.
- Dare la definizione di simbolo.
- Dare la definizione di schema.
- Spiegare il significato di alcuni schemi molto semplici che non contengono circuiti integrati.
- Elencare i simboli di un resistore, condensatore, batteria, circuito integrato, diodo e alimentatore.
- Elencare le rappresentazioni simboliche dei seguenti Outboards: LR-6, LR-7 e LR-29.
- Dire quali sono i principi base nell'uso della piastra di montaggio senza saldature e delle funzioni ausiliarie Outboards.
- Spiegare la funzione dei bus terminali di alimentazione.
- Fare alcuni esempi di funzioni ausiliarie e spiegare il loro significato.

COSA È IL BREADBOARDING?

Breadboarding vuol dire usare la piastra di montaggio per collegare temporaneamente tra loro dei circuiti elettrici; un breadboard (piastra di montaggio) è un qualsiasi mezzo ausiliario usato temporaneamente per collegare tra loro dei circuiti, per controllare la fattibilità di un circuito, di un dispositivo, di un sistema etc.

In questa trattazione userete *una solderless breadboard (piastra di montaggio senza saldature)* fabbricata dalla E & L. Instruments, Inc. (rappresentata in Italia dalla Microlem S.p.A. Milano).

PRESENTAZIONE DEL BREADBOARD

Nelle figure 9-1 e 9-2 vengono mostrate le fotografie rispettivamente del lato superiore e inferiore della piastra di montaggio SK-10. La piastra contiene 128 set di 5 terminali collegati senza saldature; due set adiacenti sono distanti 0,1 pollici in modo che i chips abbiamo spazio sufficiente. Come si può vedere dalle figure, sono divisi in due gruppi di 64 set sistemati ai lati di una stretta striscia centrale. Sui lati esterni della piastra di montaggio ci sono otto set di venticinque terminali, comunemente usati per i +0 V e i +5 V. Noi li chiameremo *bus terminali di alimentazione.*

Set di cinque terminali permettono l'alloggiamento dei chip e consentono quattro collegamenti addizionali ad ogni pin per i chip più piccoli a 14 pin e 16 pin. Di solito chiamiamo questi terminali *input/output* o (ingresso/uscita) o *terminali di I/O* (figura 9-5).

Viene usato, per questa piastra, il termine *senza saldatura* perché i collegamenti elettrici tra i componenti vengono eseguiti senza l'aiuto di un saldatore e di materiale saldante. Il sistema di collegamento usato è quello di inserire del filo elettrico spelato, di solito cavo
— 24, negli appositi contatti contenuti in ogni terminale senza saldatura. Riportiamo un brano di un articolo di *Electronics* sull'argomento.

> "I principali vantaggi di questo sistema sono:
> la velocità di collegamento senza che sia richiesta una specifica capacità, la possibilità di riutilizzare i materiali e la possibilità di poter sistemare più componenti. Siccome ogni singola parte sta in un quadretto di 0,1 pollici è molto facile trasferire il tutto su una singola piastra a circuito stampato con la stessa quadrettatura."
>
> "La piastra di montaggio va benissimo per TTL MOS e C-MOS, ma la sua capacità di tre picofarad per contatto è troppo elevata per ECL o Schottky TTL."

Noi abbiamo riscontrato un buon successo della piastra senza saldature SK-10 tra gli studenti.

USO DELLA PIASTRA SENZA SALDATURE

I dispositivi elettronici che userete sulla piastra comprendono resistori, condensatori, diodi emettitori di luce, circuiti integrati e delle *funzioni ausiliarie* dette Outboards®.
Alcuni dei dispositivi che normalmente vengono inseriti sulla piastra, sono illustrati nella figura 9-3: sono dei circuiti integrati, un condensatore e un cavo di collegamento Augat. La figura 9-4 è più completa, infatti contiene un circuito digitale completo che, in questo caso, è un contatore di sei decadi con cinque display Outboard. Queste due figure forse riescono a darvi un'idea dei possibili usi della piastra di montaggio.

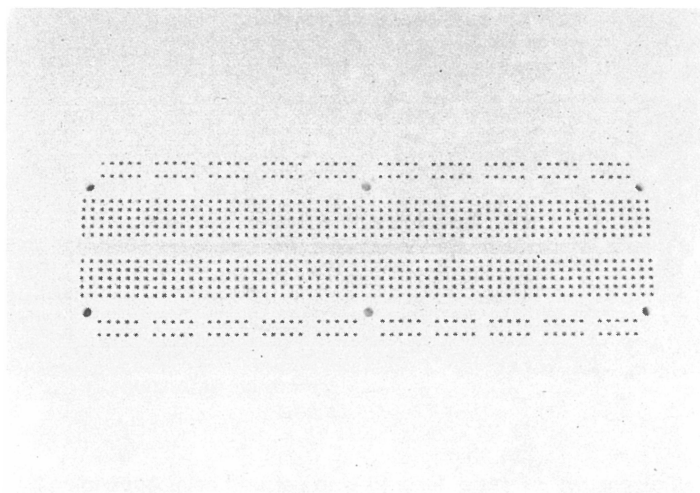



Figura 9-1. Vista superiore della piastra SK-10.

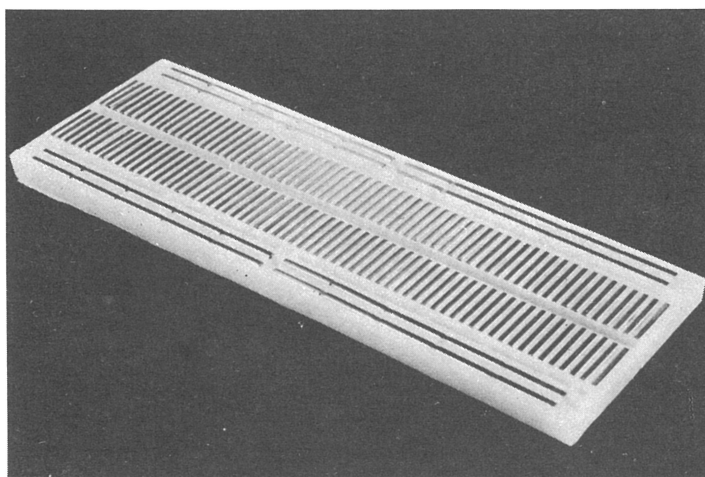



Figura 9-2. Vista inferiore della piastra SK-10.

COSA È UNA FUNZIONE AUSILIARIA?

Considereremo *funzione ausiliaria* un qualsiasi dispositivo elettronico separato necessario a rendere operativo e funzionante un circuito digitale composto di circuiti integrati, resistori, condensatori etc. collegato alla piastra.

Alcuni dispositivi che possono essere considerati funzioni ausiliarie sono:

batterie
alimentatori
switches logici
indicatori a LED
generatori di impulsi
clock
display
latch/display
contatori
generatori di funzioni
e altri

Tali dispositivi verranno descritti, uno per uno nella Appendice 3.

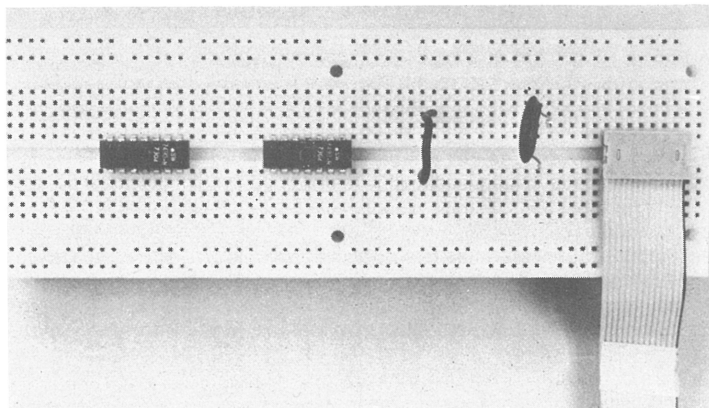
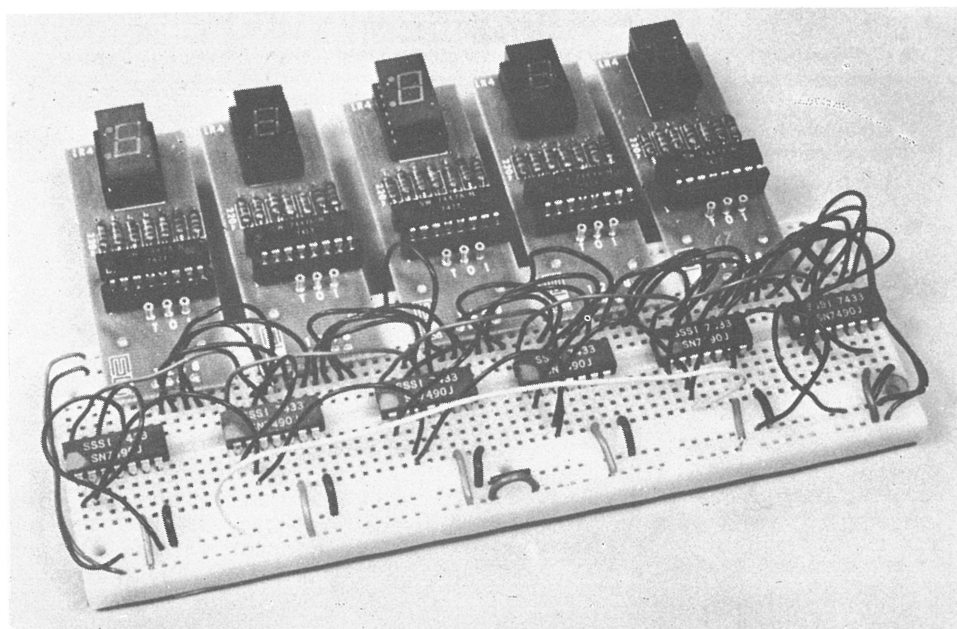



Figura 9-3. Tipici componenti elettronici che vengono montati sulla piastra SK-10.

Figura 9-4. Contatore di sei decadi montato sulla piastra SK-10 con cinque display Outboard.

COME APPLICARE L'ALIMENTAZIONE ALLA PIASTRA

Le Figure da 9-5 a 9-7 vi danno tutte le informazioni di cui avete bisogno per alimentare la piastra di montaggio SK-10. La Figura 9-5 serve a ricordarvi la configurazione base della piastra: i 128 set interni di terminali e gli 8 set esterni di venticinque terminali.

La cosa più importante che dovete ricordare è che la piastra di montaggio del microcomputer MMD-1 è diversa dalle piastre SK-10 che si trovano in commercio. Infatti, come si può vedere dalla Figura 4-4 del capitolo 4 e dallo schema alla sinistra della Figura 9-7 ci sono dei collegamenti saldati a 36 set di cinque terminali collegati elettricamente, quindi la piastra è saldata al circuito stampato ed è difficile rimuoverla. Ma quali sono i motivi per cui dovreste eventualmente toglierla? Con un uso prolungato, i terminali senza saldatura si allargano e non tengono più il contatto come quando sono nuovi. È più facile togliere i contatti da una piastra non collegata e rimpiazzare quelli vecchi che togliere tutta la piastra dal circuito stampato.

Per questa ragione vi consigliamo di fare la maggior parte degli esperimenti su una piastra separata, collegata, per mezzo dei bus terminali di +5 V e di massa, alla piastra MMD-1. Basandoci sulla esperienza fatta con alcune centinaia di studenti, crediamo che una piastra SK-10 in più, nel materiale fornito ad ogni studente, sia un'ottima cosa.

Altre cose che dovete ricordare sono:

- L'unica tensione che dovete applicare alla piastra di montaggio è +5 V, naturalmente oltre la massa (0 volt.)

- L'alimentazione può essere prelevata dall'MMD-1 e noi vi consigliamo di usare questa soluzione.

- Se dovete collocare dei circuiti digitali molto semplici, *che non devono essere usati come interfaccia al microcomputer MMD-1*, potete anche scegliere sorgenti di alimentazione diverse dall'MMD-1, come ad esempio una normale batteria.

- Collegate tutti i 100 terminali senza saldatura esterni ai +5 V. Potete farlo con dei ponticelli, come mostrato nelle Figure 9-6 e 9-7.

- Collegate tutti i 100 terminali senza saldatura interni ma lungo il bordo, col potenziale di massa. Anche questo può essere fatto con dei ponticelli come mostrato in Figura 9-6 e 9-7.

- L'alimentazione deve essere unica, preferibilmente il microcomputer MMD-1. Non cercate di alimentare un circuito digitale della piastra con due sorgenti indipendenti a +5 V.

- Infine state molto attenti quando realizzate i collegamenti di alimentazione con la piastra. Se non avete mai usato il microcomputer MMD-1 vi raccomandiamo i circuiti Outboards LR-1 o LR-25, forniti di diodi di protezione.

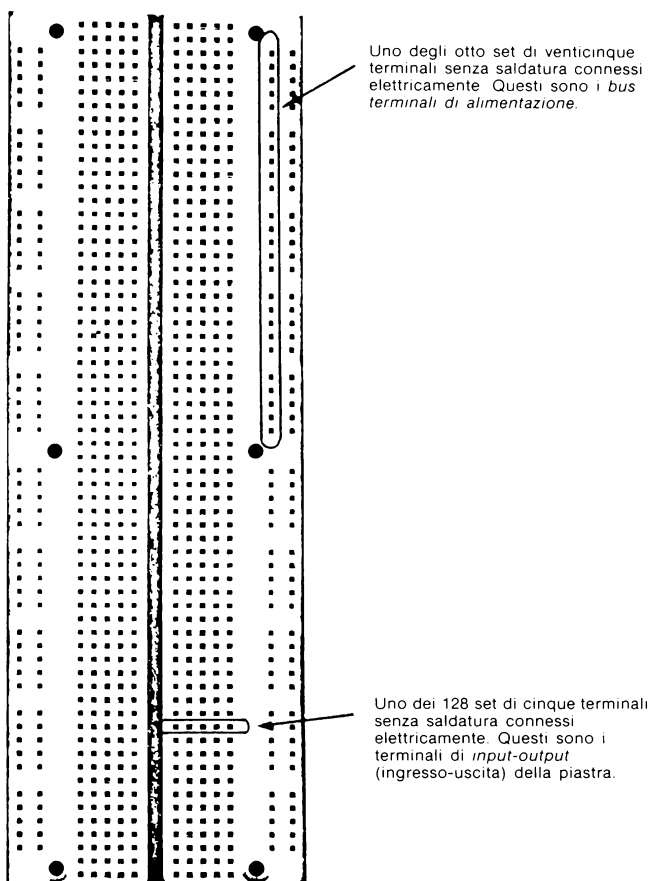



Figura 9-5. La piastra SK-10 della E. & L. INSTRUMENTS Inc. (rappresentata in Italia dalla Microlem S.p.A.).
Questa piastra universale contiene 840 contatti elastici non corrosivi, ciascuno dei quali ha una vita media di 10.000 contatti. Ci sono 128 set di cinque terminali senza saldatura connessi elettricamente posti al centro e 8 set di venticinque terminali ai bordi. Un fondo di carta sul retro della piastra protegge i set di terminali vicini da corto-circuiti causati dall'accumulo di polvere e di altri materiali conduttori.

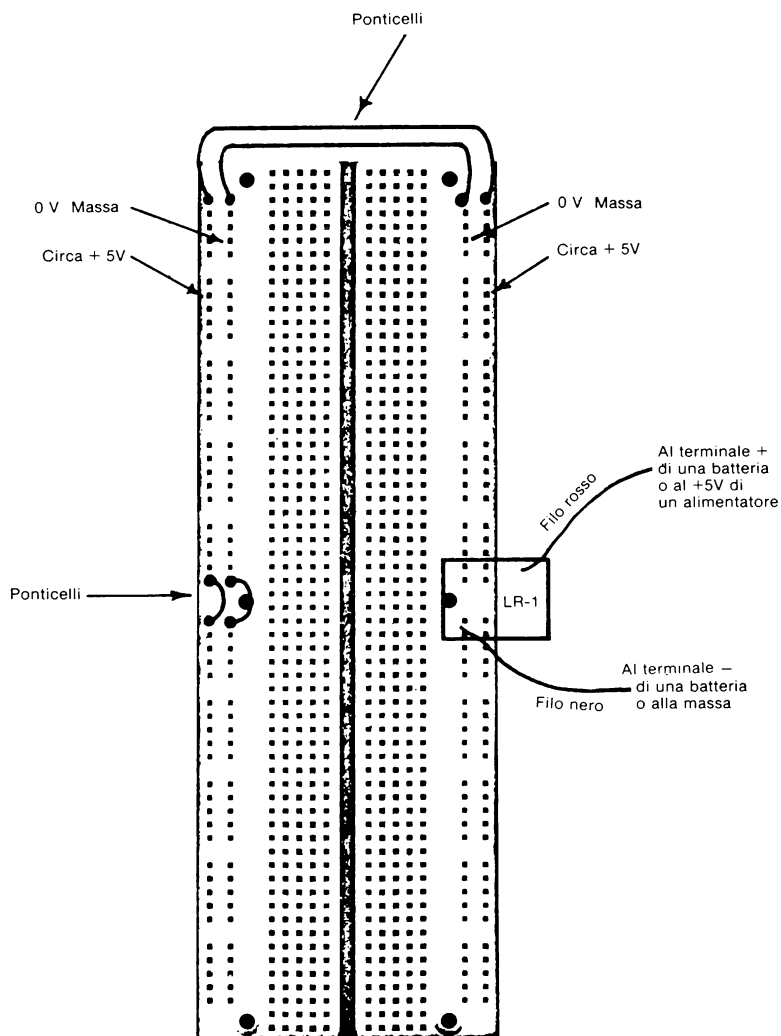



Figura 9-6. La piastra SK-10 della E.&L. Instruments, Inc. con l'Outboard LR-1 collegato in modo corretto. Sono stati anche eseguiti un paio di ponticelli che collegano tutti i terminali dei due lati della piastra. Con l'aiuto di questi ponticelli, si ottengono 100 terminali al potenziale di massa e 100 terminali al potenziale +5 V. Il potenziale più alto non deve scendere sotto i 4,75 V e non superare i 5,3 V, banda tipica di tutta la serie 7400.

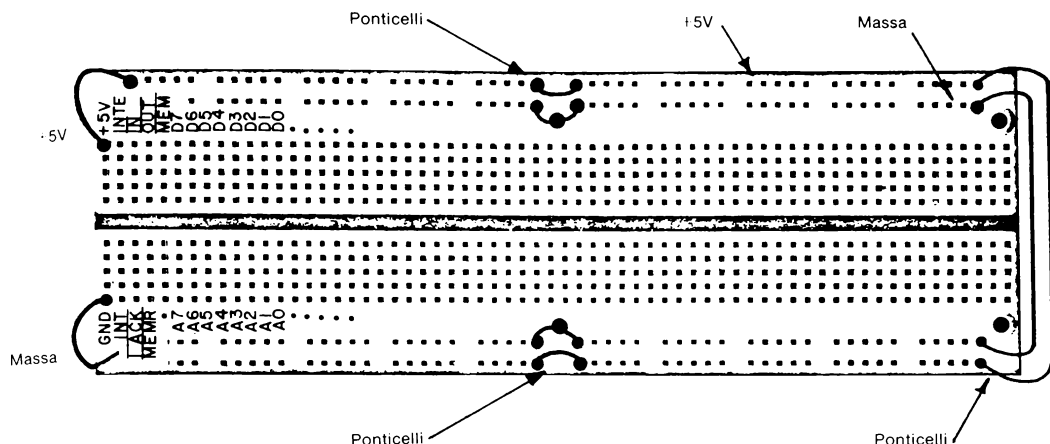



Figura 9-7. La piastra SK-10 della E. & L. Instruments, Inc. collocata sul circuito stampato del microcomputer MMD-1. Venticinque set di terminali sono collegati alla circuiteria del microcomputer; sono possibili altri undici collegamenti supplementari (dot). Questa configurazione della piastra vi permette di collegare dei circuiti di interfaccia situati sulla piastra stessa con gli indirizzi di memoria o con i bus di I/O del microcomputer. Sono altresì forniti dei segnali importanti come $\overline{\text{IN}}$, $\overline{\text{OUT}}$, $\overline{\text{MEMW}}$, $\overline{\text{MEMR}}$ e INT.

COSA È UN OUTBOARD®?

Outboard® è un marchio registrato della E. & L. Instruments, Inc. per una funzione ausiliaria montata su un piccolo circuito stampato che si collega direttamente alla piastra SK-10 e prende i +5 V dalla linea esterna dei terminali senza saldatura. Dei pin di ingresso-uscita sono collegati elettricamente ai set di cinque terminali all'interno della piastra di montaggio. Si possono facilmente vedere le caratteristiche di un Outboard dalla figura A-1, che è una vista posteriore dell'Outboard LR-7, generatore d'impulsi doppio (Dual pulser).
L'Outboard ha dei collegamenti con la piastra SK-10 in sei punti: + 5 V sul bus esterno di terminali di alimentazione, 0 V sul bus interno di terminali di alimentazione, e quattro set di *terminali di input-output* (ingresso/uscita). Per fare questi collegamenti è sufficiente esercitare una leggera pressione sull'Outboard verso la piastra dopo averlo collegato nella giusta posizione lungo i bordi.

I vantaggi della concezione degli Outboard è che sono "compatti", facili da riparare se danneggiati, portatili, e possono essere messi praticamente in qualsiasi parte della piastra SK-10. Ogni singolo Outboard è descritto nella.Appendice 3.

SIMBOLI E SCHEMI

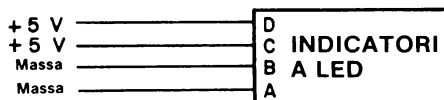
L'Oxford English Dictionary definisce un *simbolo* come "un carattere o un segno usato per rappresentare qualcosa; una lettera, figura o segno che rappresenta convenzionalmente un qualche oggetto, processo etc.".
In elettronica, i simboli vengono usati per rappresentare i componenti quali resistori, condensatori, batterie, circuiti integrati, diodi, transistori, diodi emettitori di luce (LED) e cosí via. In questa trattazione, useremo i simboli comunemente in uso per i componenti elettronici, e useremo anche dei simboli speciali che rappresentano le funzioni ausiliarie che abbiamo precedentemente descritto. Questi simboli delle funzioni ausiliarie sono stati introdotti nei Bugbooks I e II.

Uno *schema* è una rappresentazione grafica di componenti elettronici e di funzioni ausiliarie che sono collegate per comporre un dato circuito elettronico; in queste rappresentazioni vengono usati i simboli. L'esperienza ha dimostrato che gli schemi sono il modo più semplice e rapido per trasmettere le informazioni necessarie. "Una fotografia vale cento parole". I simboli sono una specie di stenografia elettronica; per mezzo dei simboli, un circuito elettronico digitale può essere assemblato in un tempo brevissimo partendo dal solo schema. Generalmente, l'uso dei simboli permette di dare tutte le informazioni necessarie in uno schema relativamente piccolo, cioè in uno schema la cui grandezza dipende dalla complessità del circuito. Inoltre uno schema è di più facile comprensione di una fotografia del circuito assemblato ed infine, siccome i simboli e gli schemi dei circuiti sono ormai standardizzati, possono essere facilmente interpretati.

Nelle pagine successive vengono descritti molti dei simboli più comunemente usati (tabella 9-1).

ALCUNI SCHEMI SEMPLICI

In questo capitolo farete esperienza nel collegare alcuni semplici circuiti che usano gli Outboards. Gli schemi che dovrete seguire sono tra i più semplici.

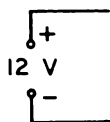
Per prima cosa dovrete mostrare l'uso degli schemi indicatori a LED per determinare gli stati logici 0 e 1. Il circuito è:

Quando l'alimentazione è correttamente applicata alla piastra, gli indicatori a LED A e B sono spenti mentre C e D sono accesi e se, per caso, avete eseguito un errato collegamento, i LED non si accendono. Ricordate che sono necessari solo +5 V e massa.

Nel primo circuito collegherete quattro switch logici ai quattro indicatori a LED

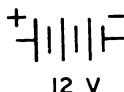
Sommario dei simboli per componenti elettronici e funzioni ausiliarie.

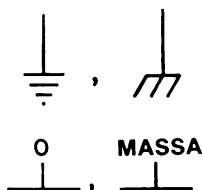
SORGENTE DI TENSIONE

Una sorgente di tensione o alimentazione sarà indicata da questo simbolo e si supporrà riferita alla massa se non altrimenti indicato.

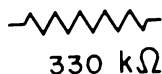
ALIMENTAZIONE

Una alimentazione o sorgente di tensione rappresentata in un altro modo. In questi manuali, l'alimentazione sarà indicata da questo simbolo. La polarità e la tensione saranno di solito forniti.

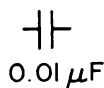
BATTERIA

Questo è il simbolo normale per una batteria ad un elemento. La stanghetta più breve rappresenta il polo negativo, mentre la più lunga rappresenta il polo positivo. I segni (−) e (+) possono mancare. Qualche volta è data anche la tensione.

Una batteria a più elementi sarà rappresentata con più stanghette. Di solito è dato anche la tensione.

MASSA

Questi simboli indicano la massa, che alcune volte è abbreviata con GRD o GND (GROUND). Tutti i punti di un circuito con questo simbolo devono essere considerati allo stesso potenziale elettrico. Generalmente questo potenziale è considerato pari a 0 V.

Anche questi simboli indicano la massa e saranno spesso usati in questo manuale.

RESISTORE

Questo simbolo indica un resistore di valore costante. Di solito il valore del resistore è dato in ohm (Ω),kiloohm (kΩ) o megaohms (MΩ) dove kilo=1000 e mega = 1.000.000.

CONDENSATORE

Questo simbolo rappresenta un condensatore di valore costante. Di solito il valore del condensatore è dato in microfarad (μF), nanofarad (nF) o picofarad (pF) dove micro=10^{-6}, nano = 10^{-9} e pico = 10^{-12}

DIODO

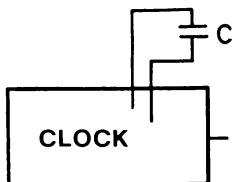
Questo simbolo rappresenta un diodo. La direzione della freccia e la stanghetta indicano la polarità. Una corrente significativa passa attraverso il diodo solo quando è polarizzato direttamente, con l'*anodo* collegato al potenziale positivo rispetto al catodo, come è illustrato nella figura.

SWITCH LOGICI

Questo simbolo rappresenta un gruppo di quattro switch logici ed è una funzione ausiliaria di capitale importanza in quasi tutti i circuiti digitali. Le lettere A B C D rappresentano quattro bit di un *parola binaria* di *4-bit,* dove A è il *bit meno significativo* e D il *più significativo.*

INDICATORI A LED

Questo simbolo rappresenta un gruppo di quattro indicatori a LED che costituisce una funzione ausiliaria di capitale importanza in tutti i circuiti digitali. Come per gli switch logici, le lettere ABCD rappresentano i quattro bit di una parola binaria di 4-bit.

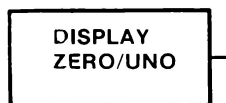
CLOCK

Questo simbolo rappresenta un clock, funzione ausiliaria che produce un'onda quadra da circa +5V a 0V. Di solito il valore della frequenza del clock viene dato in Hertz (Hz) o Kilohertz (kHz) dove un Hertz (Hz) è la frequenza corrispondente ad un ciclo/secondo. Una serie di *impulsi di clock* prodotta appunto da un clock, viene di solito detta *treno di impulsi* di clock.

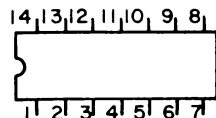
GENERATORE DI IMPULSI
(Pulser)

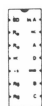
Questo simbolo rappresenta un generatore di impulsi, una funzione ausiliaria che vi permette di dare un singolo *impulso* ad un circuito elettronico digitale. Le due uscite, o output, sono una il complemento dell'altra; un'uscita è 1 quando l'altra è 0 e viceversa. Se non viene indicato altrimenti, il generatore di impulsi è debounced, termine che spiegheremo più avanti.

DISPLAY A LED A SETTE SEGMENTI
(Seven-Segment Led Display)

Questo simbolo rappresenta un display a diodi emettitori di luce (LED) a sette segmenti, funzione ausiliaria che ha la capacità di visualizzare sia dei numeri decimali da 0 a 9 che un gruppo di sei simboli che rappresentano i numeri decimali 10, 11, 12, 13, 14 e 15. Come nel caso degli switch e degli indicatori a LED, le lettere A B C D rappresentano i quattro bit della parola binaria di 4 bit.

DISPLAY ZERO/UNO

Questo simbolo rappresenta un display zero/uno, funzione ausiliaria fatta di un display a LED a sette segmenti che può visualizzare il numero decimale 0 o 1 a seconda se l'ingresso è rispettivamente di 0 o +5 V.

CIRCUITO INTEGRATO
(Integrated Circuit Chip)

Questo è uno dei diversi, ma simili, simboli che vengono usati per rappresentare un circuito integrato.
Due sono le informazioni date dal simbolo: il sistema di numerazione dei pin e la funzione del circuito integrato corrispondente a ciascuno dei diversi pin che esistono.

BUGBACK®*

LR BUGBACK® è un simbolo che viene affisso sul retro dei circuiti integrati per evitare di dover ripetutamente guardare le specifiche del costruttore riguardo ai pin e alle funzioni. Il BUGBACK® riportato qui a sinistra è quello del contatore decimale 7490.

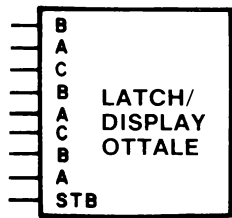
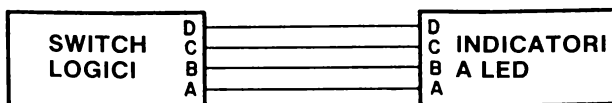
TIMER PROGRAMMABILE

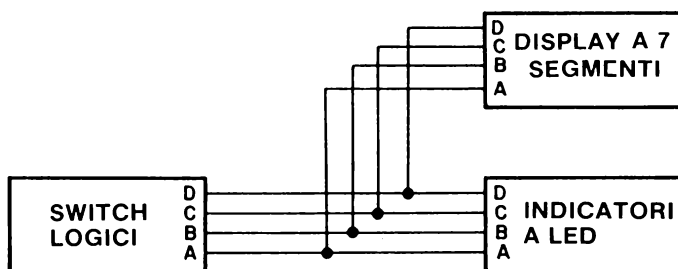
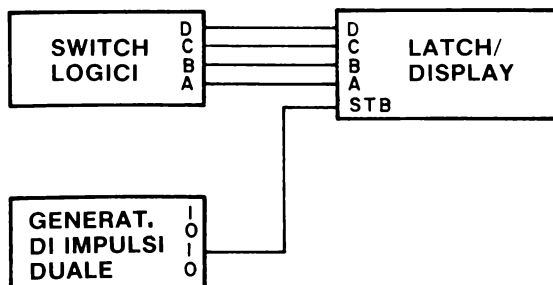
Questo simbolo rappresenta l'uscita di un timer programmabile, nel quale le frequenze di clock possono essere generate in multipli della frequenza base per 1,2,4,8,16,32,64 o 128.
L'uscita "0" è la frequenza più bassa e la "7" è la più alta.

* BUGBACK® è un marchio registrato.

LATCH/DISPLAY

Questo simbolo rappresenta un display a LED decimale o esadecimale che contiene al suo interno un latch a quattro bit, cioè una memoria a quattro bit. Il display decimale ha un'uscita che va da 0 a 9, mentre quello esadecimale ha un'uscita che va da 0 a 9 e da A a F in modo da rappresentare i sedici stati esadecimali.

LATCH/DISPLAY OTTALE

Questo simbolo rappresenta un latch/display a tre digit ottali che traduce gli otto bit di ingresso in numeri ottali da 000_8 a 377_8. I tre latch contenuti al suo interno sono collegati tra di loro in modo da poter parallelizzare una parola di otto bit.

Mettendo ogni switch al livello 0 o 1, potete stabilire quale LED si accende. Se tutti gli switch logici, ABCD, sono al livello 1, allora si accenderanno tutti i LED ABCD, poiché esiste una corrispondenza uno a uno tra gli stati logici di ogni switch e lo stato logico dei LED collegati.

Nel terzo circuito,

aggiungete un display a sette segmenti e confrontate gli indicatori a LED con quello che visualizza il display. Osserverete una corrispondenza tra l'uscita binaria degli indicatori a LED e l'uscita decimale del display. Provando tutte le sedici combinazioni possibili degli switch logici potete comporre la tabella della verità. Se non avete un display a sette segmenti, potreste collegare il seguente circuito

che impiega l'Outboard latch/display LR-29 e un generatore di impulsi. Per trasmettere l'informazione al display dovete premere e rilasciare il generatore d'impulsi e se volete che il latch/display mostri tutti i cambiamenti degli switch logici, dovete collegare l'ingresso STB all'uscita "0" del generatore d'impulsi.

REGOLE PER ESEGUIRE GLI ESPERIMENTI

Ormai avete già fatto un certo numero di esperimenti di programmazione sul microcomputer MMD-1, partendo da questa unità, comincerete a fare esperimenti nei quali collegherete dei circuiti digitali sulla piastra di montaggio SK-10. I primi esperimenti verranno eseguiti con funzioni ausiliarie e circuiti integrati, poi *interfaccerete* questi circuiti col microcomputer usando i terminali senza saldatura mostrati alla sinistra della Figura 9-7. Vi consigliamo di osservare le seguenti regole base prima di cominciare ogni esperimento.

1. Se l'esperimento serve ad illustrare un concetto di elettronica digitale e non richiede alcun interfacciamento col microcomputer, *usate la piastra di montaggio SK-10 non collegata.* Così facendo non occupate la piastra dell'MMD-1 e la lasciate libera per chi volesse costruire un circuito d'interfaccia.

2. Pianificate in precedenza l'esperimento, cioè cercate di conoscere in anticipo i risultati che vi aspettate di osservare.

3. **IMPORTANTE:** Prima di usare la piastra di montaggio staccate il collegamento +5 V dal bus esterno della piastra; questo accorgimento serve a prevenire i cortocircuiti accidentali sulla piastra. Se invece state collegando un circuito al microcomputer non vi chiediamo di togliere l'alimentazione all'intero microcomputer perchè così facendo cancellereste tutta la memoria a lettura/scrittura. *Inoltre togliete dalla piastra di montaggio tutti i circuiti e i fili che non sono necessari.*

4. Quando la connessione a +5 V tra l'alimentazione e la piastra è stata staccata, allora collegate, ma con attenzione, i circuiti digitali. Per prima cosa, prima di *ogni altro, collegate l'alimentazione ad ogni circuito integrato.* Abbiamo notato infatti che il 60% di tutti gli errori di collegamento riguardano l'alimentazione e il modo migliore di danneggiare un circuito è quello di sbagliare a collegarlo alla sorgente di alimentazione.

5. Prestate molta attenzione al modo di disporre i singoli chip e le funzioni ausiliarie sulla piastra di montaggio, infatti una disposizione ben curata riesce spesso a minimizzare la "giungla" di fili abbastanza tipica in ogni circuito digitale anche di poca complessità.

6. Controllate i circuiti che avete collegato per essere certi che tutto sia stato fatto in modo corretto. *Fate molta attenzione ai collegamenti di alimentazione di ogni singolo circuito integrato.* Se non sono corretti potreste rovinare i chip e probabilmente cancellare la memoria a lettura/scrittura. Alcuni chip non bruciano subito; toccate con un dito la superficie superiore dei chip di cui non siete proprio sicuri e se è caldo c'è qualcosa di sbagliato.

7. Collegate i +5 V solo quando tutto è stato controllato. Adesso potete usare il test "del dito" per vedere se qualche chip è troppo caldo. Fate qualche test iniziale per controllare rapidamente se ogni singolo circuito è collegato nella maniera voluta.

Se tutto è a posto proseguite l'esperimento, ma se qualcosa non va, togliete immediatamente la tensione e cercate gli errori di collegamento.

8. *Non scollegate i circuiti fino a che non avete finito l'esperimento* e controllate l'esperimento successivo per vedere se i circuiti che avete già collocato servono ancora o no.

9. Quando siete al termine della giornata di lavoro, staccate il collegamento di alimentazione principale dal microcomputer e dalla piastra di montaggio separata.

Queste regole sono già state illustrate nel capitolo 4.

COME SONO PRESENTATE LE ISTRUZIONI PER GLI ESPERIMENTI

Le istruzioni per ogni esperimento saranno presentate nel formato descritto qui sotto:

Scopo

Quello che sarà detto sotto questo titolo servirà a definire lo scopo dell'esperienza ed è quindi meglio che lo teniate presente nella conduzione dell'esperimento.

Configurazioni dei pin dei circuiti integrati

Le configurazioni dei pin, date col permesso della TEXAS INSTRUMENTS INCORPORATED e della INTEL CORPORATION, vengono fornite sotto questo titolo per tutti i circuiti integrati usati negli esperimenti. Questo paragrafo può mancare se la configurazione è identica a quella dell'esperimento precedente.

Schema del circuito

Vi sarà fornito lo schema completo del circuito che dovrete collegare durante l'esperimento e dovrete analizzare questo schema per capire il circuito prima di passare alla parte realizzativa. Controllate tutti i numeri dei pin di tutti i collegamenti al circuito integrato e ricordate che non sono segnati i collegamenti di alimentazione alle porte. Fate attenzione ai collegamenti di +5 V e massa: il circuito non funzionerà in caso di errori.

Programma

Se l'esperimento comprende la programmazione o l'interfacciamento del microcomputer, vi sarà fornito il programma da caricare in memoria all'indirizzo indicatovi. Vi suggeriamo di collegare prima i circuiti di interfaccia e di provarli, se possibile, prima di caricare il programma nella memoria a lettura/scrittura. Se non verrà diversamente indicato, il programma dovrà essere caricato all'indirizzo di memoria LO della prima istruzione, mentre l'indirizzo HI sarà di solito 003_8. Per sfruttare vantaggiosamente il tasto RESET, è possibile mettere una istruzione JMP a HI=003_8 e LO=000_8; comunque discuteremo in seguito l'uso, a questo scopo, della istruzione JMP.

Passi

In relazione ad ogni passo sequenziale, cioè passo 1, passo 2,......., compariranno le istruzioni dettagliate di quello che dovrete fare in quella particolare sezione di esperimento. Durante l'esperimento vi verrano poste anche delle domande cui voi dovrete rispondere, negli appositi spazi, mentre eseguite l'esperimento. Dopo aver scritto le risposte, controllatene l'esattezza confrontandole con le risposte che compariranno nel testo subito dopo. Se ce ne fosse la necessità e se le due risposte non coincidessero, ve ne accorgereste sicuramente (correggete la risposta se vi è possibile) prima di continuare col resto dell'esperimento.

Domande

Vi saranno frequentemente poste delle domande per provare (a) quanto avete capito dell'esperimento che avrete appena terminato, (b) la vostra abilità nell'anticipare esperimenti e problemi futuri, (c) la vostra capacità di mettere in relazione il materiale fornitovi dal testo con le nozioni apprese sperimentalmente, oppure (d) la vostra abilità nel trarre conclusioni o nel fornire spiegazioni plausibili a problemi più generali riguardanti del materiale che non vi sia stato precedentemente spiegato.

ALCUNI CONSIGLI UTILI

Vi saranno forniti un certo numero di consigli sulla scelta e sull'uso di circuiti integrati, componenti elettronici, fili, piastre, funzioni ausiliarie e simili. Periodicamente ripeteremo i consigli più importanti.

Cavi e fili

- I fili che possono essere usati sulla piastra di montaggio SK-10 sono di tre grandezze,

#22
#24
#26

- Noi, dopo un'esperienza di almeno due anni, preferiamo il filo #24 e, in seconda scelta del filo #22.
- Gli estremi del filo possono essere spellati per una lunghezza di 5÷8 mm; infatti queste due lunghezze sono le migliori per la piastra SK-10.
- Il cavo per i collegamenti elettrici sulla piastra deve essere rigido e non flessibile e del tipo a treccia.
- Per i collegamenti che si fanno normalmente su una piastra servono cavi di almeno quattro o sei colori diversi. Vi consigliamo di usare i fili rossi e neri per i collegamenti di breve lunghezza, colori standard per i +5 V e la massa.

Piastra (Breadbording) senza saldature

- Vi raccomandiamo di fare molta attenzione nell'uso dei colori e delle lunghezze dei fili quando collegate dei circuiti digitali abbastanza complessi, infatti esistono molti modi per realizzarli ed è lasciato alla vostra fantasia di scegliere il "migliore".
- Il gruppo di 100 terminali più esterni, verso il bordo della piastra, è quello dei +5 V. Assicuratevi che tutte le sezioni siano collegate tra loro.
- Il gruppo di 100 terminali più all'interno, ma sempre sul bordo della piastra, è il bus dei terminali di massa. Assicuratevi che anche qui tutte le sezioni siano collegate tra di loro.
- Non inserite mai nei terminali della piastra dei fili, cavi pin o dei dispositivi troppo grossi.
- Per prima cosa collegate ai circuiti integrati l'alimentazione.
- Non inserite mai nei terminali dei fili piegati, ma raddrizzatene le estremità con una pinza e se l'estremità è irrecuperabile, tagliatela e spelate il filo per circa 6 mm.
- Dislocate sulla piastra i circuiti integrati e le funzioni ausiliarie in modo che i collegamenti necessari risultino i più corti possibile.
- Evitate di costruire una "giungla" di fili di colore e lunghezza troppo fantasiosa.
- Cercate di usare fili della minor lunghezza possibile, cioè limitate i fili in modo da poter vedere il meglio possibile i pin dei circuiti integrati.

Funzioni ausiliarie Outboards

- Manipolatele con cautela.
- Per togliere una funzione ausiliaria Outboard dalla piastra SK-10, usate un utensile di legno o uno strumento di plastica. Uno strumento abbastanza buono potrebbe essere un cacciavite con la punta ricoperta di nastro adesivo per minimizzare il pericolo di danneggiare il circuito stampato sul lato inferiore dell'Outboard.
- Non mettete un circuito integrato sul lato opposto in corrispondenza dei pin di un Outboard, potrebbero succedere degli inconvenienti al chip all'Outboard o ad entrambi.
- Mettete invece gli Outboard uno opposto all'altro sull'SK-10 per risparmiare più spazio possibile.
- Maneggiate gli Outboard con cura. Per esperienza abbiamo visto che i punti più delicati sono i pin dei +5 V e di massa; entrambe hanno la tendenza a piegarsi o spezzarsi se vengono trattati rudemente.

- La nomenclatura sugli Switch logici LR-2 e sugli Outboard indicatori a LED LR-6 è arbitraria e i quattro pin di ingresso o uscita possono essere chiamati 0123, 3210, ABCD, DCBA. 1248 o in qualsiasi altro modo.

- Ricordate di premere gli switch del generatore di impulsi sull'Outboard LR-7, e non di rilasciarli.

Nei capitoli successivi vi saranno forniti altri consigli sui componenti, sui dispositivi elettronici e sui circuiti integrati.

INTRODUZIONE AGLI ESPERIMENTI

I seguenti esperimenti servono a farvi conoscere gli Outboard più importanti che poi userete nella costruzione di circuiti digitali; farete inoltre esperienza nell'uso della piastra di montaggio e nella lettura ed interpretazione degli schemi. Vi raccomandiamo di fare questi esperimenti su una piastra di montaggio SK-10 separata e non su quella collegata al circuito stampato dell'MMD-1.

Gli esperimenti che farete possono essere cosí riassunti:

Esperimenti N.	Commento
1	Mostrare l'uso dell'indicatore a LED Outboard LR-6 per determinare gli stati logici 1 e 0.
2	Ripetizione dell'esperimento N. 1, nel quale un Outboard di alimentazione LR-1 viene usato per fornire tensione ai set di terminali interni ed esterni.
3	Dimostrare cosa succede quando si collegano in modo sbagliato le prese a coccodrillo dell'Outboard di alimentazione alla sorgente di alimentazione.
4	Dimostrare l'uso del LR-2 logic switch Outboard per generare stati logici 0 e 1.
5	Dimostrare come lavora l'LR-4 seven segment LED display Outboard per visualizzare parole di 4 bit.
6	Dimostrare come lavora L'LR-4 usato come display zero/uno.
7	Dimostrare come opera l'LR-7 dual pulser Outboard.
8	Confrontate le uscite "0" e "1" di un Outboard LR-7.

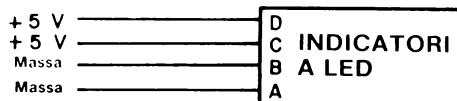
Questi otto esperimenti sono quelli fondamentali di questo capitolo; potete usare il microcomputer MMD-1 come sorgente di alimentazione. In più aggiungiamo altri due esperimenti basati sull'uso degli Outboard LR-25 e LR-29.

9	Dimostrare come funziona l'Outboard LR-25 Breadboarding station
10	Dimostrare come funziona l'LR-29 latch/display Outboard o la funzione equivalente svolta dall'Outboard LR-25, Breadboarding station.

ESPERIMENTO N. 1

Scopo

Lo scopo di questo esperimento è di mostrare l'uso dell'Outboard LR-6 per determinare gli stati logici 0 e 1.

Schema del circuito

Passo 1

Studiate la piastra di montaggio illustrata nella Figura 9-8. Fate attenzione alla alimentazione, all'uso dei ponticelli per dare tensione a tutti i cento bus terminali a +5 V a tutti i cento bus terminali di massa e ai fili collegati con l'Outboard LR-6.

Passo 2

Inserite piano piano l'Outboard LR-6 sulla piastra di montaggio, nel modo illustrato nella Figura 9-8.

Passo 3

Collegate i ponticelli che non verrano mai più tolti. Collegate poi i quattro ingressi all'Outboard LR-6 sia ai +5 V che alla massa, come indicato nello schema. *Vi raccomandiamo di non fare i collegamenti dei ponticelli di +5 V, in alto a sinistra nella Figura 9-8, prima di aver correttamente collegato l'Outboard e gli altri ponticelli.* In altre parole non collegate l'alimentazione alla piastra di montaggio prima che tutto il circuito non sia completamente collegato. Se seguite questa procedura ridurrete al minimo la possibilità di danneggiare sia l'Outboard che i circuiti integrati, che poi dovrete usare anche negli esperimenti dei successivi capitoli.

Passo 4

Collegate l'alimentazione a +5 V al bus di terminali più esterni facendo le connessioni mostrate nella Figura 9-8 in alto a sinistra sull'angolo della piastra. Dovreste vedere che due dei quattro indicatori a LED si accendono; per la precisione quelli collegati con i +5 V. Il fatto che siano accesi significa che sono allo stato logico 1.

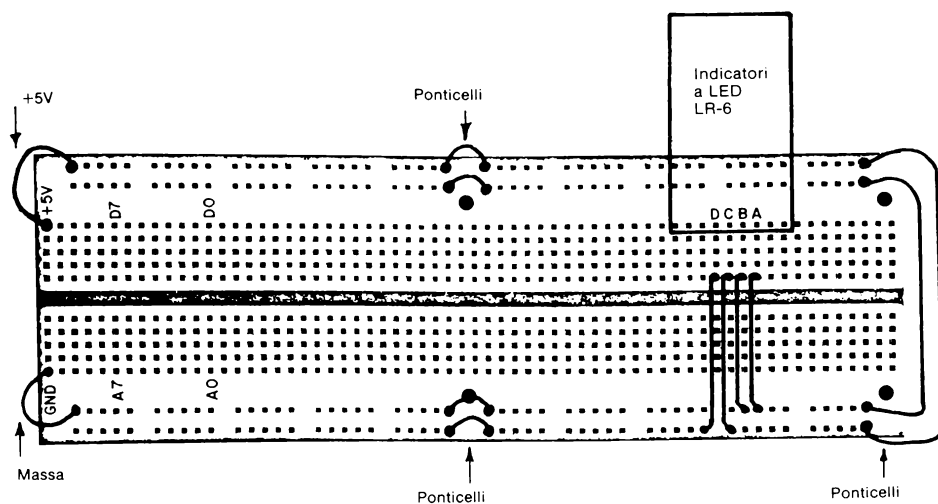



Figura 9-8. Schema dei collegamenti per l'Esperimento N. 1.
Il collegamento più importante è quello dei bus di terminali in alto a sinistra. Non eseguite questa connessione prima di aver accertato che tutti gli altri collegamenti della piastra di montaggio siano esatti.

Passo 5

Osserverete che le luci A e B sono spente. A che stato logico corrisponde un indicatore a LED spento?

Allo stato logico 0.

Passo 6

Studiate attentamente lo schema del circuito di questo esperimento e disegnate uno schema in cui tutte e quattro le luci, da A a D, siano allo stato logico 1. Fate questo circuito sulla piastra di montaggio e dimostrate che funziona correttamente. *Quando cambiate i collegamenti del circuito, Vi raccomandiamo di staccare i +5 V mostrati nella figura 9-8 all'angolo in alto a sinistra della piastra di montaggio.* Dopo aver ricollegato il circuito, ripristinate il collegamento dei +5 V.

Passo 7

Come potete stabilire, per mezzo di un'unico LED, se un terminale elettrico, un pin di un circuito integrato, un filo o un resistore, etc. di un circuito digitale sono allo stato logico 1 o 0?

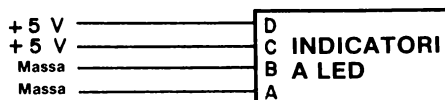
Dovete collegare un singolo indicatore a LED, pin, resistore, etc. e se la luce si accende, lo stato è 1, se invece la luce rimane spenta, lo stato è 0.

Quando avete terminato questo esperimento, staccate la tensione di +5 V dal circuito e passate all'esperimento successivo.

ESPERIMENTO N. 2

Scopo

Lo scopo di questo esperimento è dimostrare come sia possibile eseguire l'Esperimento N. 1 in assenza della piastra di montaggio del microcomputer MMD-1. *Considerate facoltativo questo esperimento se avete già fatto l'Esperimento N.1.*

Schema del circuito

Passo 1

Se avete già fatto l'Esperimento N. 1 e non avete un Outboard di potenza LR-1, passate direttamente all'Esperimento N.4.

Passo 2

Studiate lo schema del circuito mostrato in Figura 9-9; noterete subito che è molto simile a quello della Figura 9-8. Il vantaggio dell'Outboard di potenza LR-1 è che non è possibile sbagliare i collegamenti col bus dei terminali di alimentazione esterni come nel caso della Figura 9-8. Un altro vantaggio dello schema mostrato in Figura 9-9 è dato dal fatto che è possibile fare l'esperimento senza il microcomputer. Se volete fare l'esperimento su una piastra SK-10 separata dovete far riferimento ai Bugbook I e II.

Passo 3

Inserite dolcemente gli Outboard LR-1 e LR-6 sulla piastra di montaggio come mostrato in Figure 9-9 e 9-10. Non collegate le prese a coccodrillo dell'Outboard LR-1 all'alimentazione, prima dovete fare degli altri collegamenti.

Passo 4

Collegate i ponticelli nel modo indicato in figura 9-9. Quelli resteranno collegati anche dopo aver tolto l'Outboard LR-6.

Passo 5

Collegate i quattro ingressi dell'Outboard LR-6 ai +5 V o alla massa.

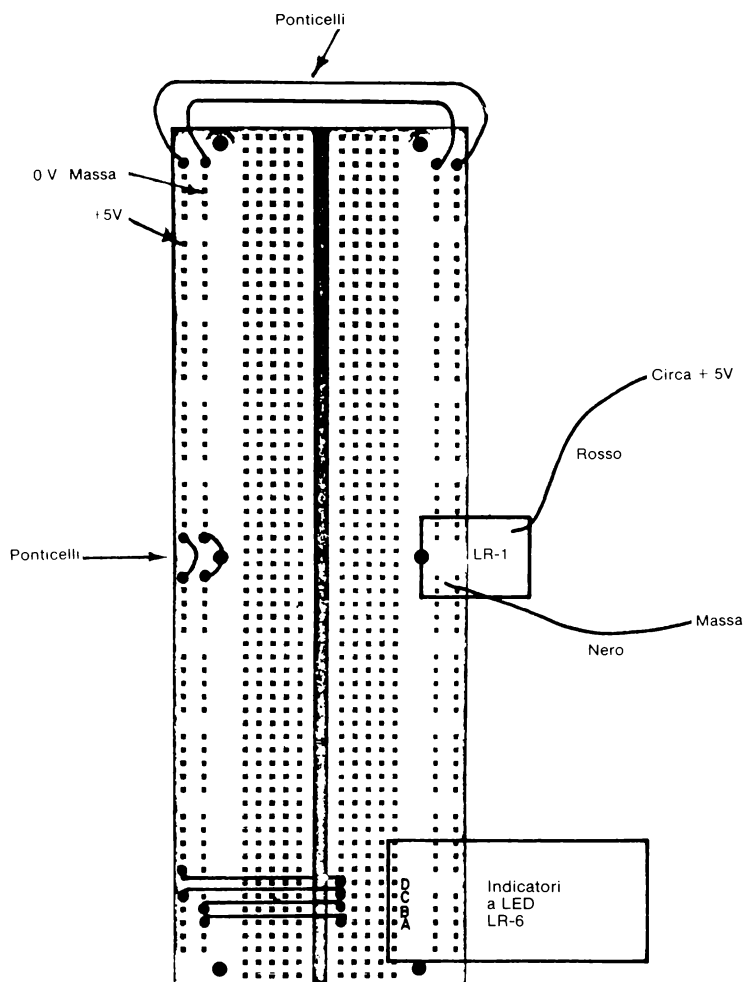



Figura 9-9. Schema dei collegamenti per l'Esperimento N. 1.
Le prese a coccodrillo possono essere collegate sia ad una batteria di 6 V che a una alimentazione di +5 V. Il potenziale del filo rosso dell'Ourboard di alimentazione LR-1 non deve essere inferiore a +4,75 V come mostrato nello schema.

Passo 6

Collegate le due prese a coccodrillo ad una batteria, come mostrato in Figura 9-10, oppure ad un qualsiasi altro tipo di alimentazione. Osserverete che due dei quattro indicatori a LED, si accendono.

Passo 7

Per gli indicatori a LED dell'LR-6, una luce accesa corrisponde allo stato logico 1 e una luce spenta corrisponde allo stato logico 0. Quali indicatori sono accesi e quali sono spenti?

C e D sono accesi; A e B sono spenti.

Passo 8

Quali indicatori a LED sono allo stato logico 1?

Gli indicatori a LED C e D.

Quali sono allo stato logico 0?

Gli indicatori a LED A e B.

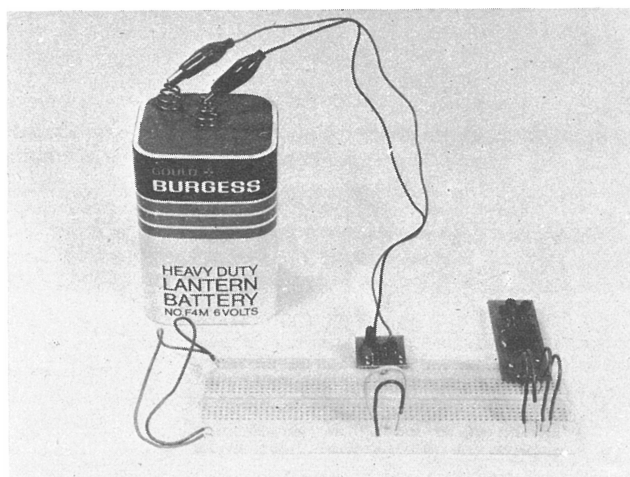



Figura 9-10. Esperimento N. 2

Passo 9

Disegnate uno schema nel quale tutti i quattro indicatori a LED, da A a D, sono allo stato logico 0. Collegate questo circuito sulla piastra e dimostrate che funziona.

Passo 10

Una volta terminato questo esperimento, staccate le prese a coccodrillo. Adesso potete scegliere tra due modi di fare dei semplici circuiti:

1. Sulla piastra attaccata al microcomputer

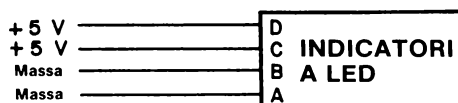
2. Sulla piastra separata, ammesso che abbiate un Outboard di potenza LR-1 o che ponticelliate la piastra attaccata al microcomputer con quella separata.

Noi vi consigliamo di fare più esperimenti possibili sulla piastra separata. *Staccate sempre l'alimentazione della piastra quando pensate di cambiare un circuito digitale o di aggiungerne uno nuovo.*

ESPERIMENTO N. 3

Scopo

Lo scopo di questo esperimento è di mostrare cosa succede quando si collegano in modo sbagliato le prese a coccodrillo dell'Outboard di alimentazione LR-1 ad una batteria.

Schema del circuito

Passo 1

Prima di iniziare questo esperimento dovete aver già completato il N. 1; infatti lo schema del circuito è il medesimo.

Collegate il filo nero e la pinza a coccodrillo relativa al terminale + o +6 della batteria o al terminale di +5 V della sorgente di alimentazione. Collegate il filo rosso e la presa a coccodrillo relativo al terminale — della batteria o della alimentazione. Scrivete qui sotto se gli indicatori C e D sono accesi o spenti.

Nel nostro caso, sono spenti.

Passo 2

Scambiate tra loro i collegamenti delle prese a coccodrillo sulla batteria o sulla alimentazione. Il filo rosso sarà quindi sul terminale + e il filo nero sul terminale —.
I LED C e D si sono accesi?

Nel nostro caso si sono accesi. I LED A e B erano sempre rimasti accesi perché abbiamo dimenticato di scambiare i collegamenti fatti nel Passo 7 dell'Esperimento N. 2. Infatti anche quando avevamo collegato questo circuito nel modo corretto, solo C e D erano accesi.

Passo 3

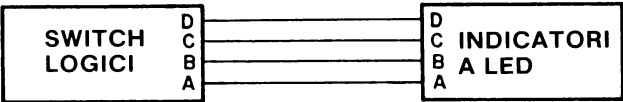
Il LED dell'Outboard di alimentazione LR-1 è acceso o spento? in che modo potete determinare se l'Outboard LR-1 è collegato senza errori all'alimentazione?

Se l'Outbord LR-1 è collegato nel modo esatto, il LED è acceso, come nel nostro caso.

ESPERIMENTO N. 4

Scopo

Lo scopo di questo esperimento è di mostrare l'uso dell'LR-2 logic switch Outboard per generare stati logici 0 e 1.

Schema del circuito

Passo 1

Ogni volta che collegate un nuovo circuito digitale, dovete sempre staccare i +5 V di alimentazione dal bus di terminali più esterno; ci aspettiamo che lo facciate per ognuno degli esperimenti di questo capitolo. Non collegate mai un nuovo circuito, nè smantellatene uno vecchio con l'alimentazione collegata alla piastra di montaggio.

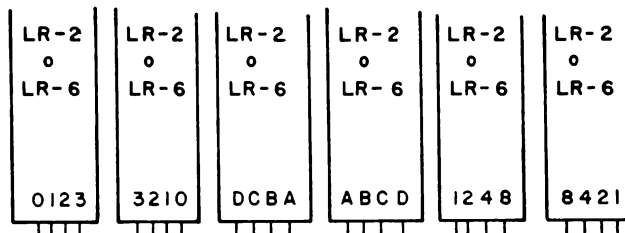
Staccate i +5 V dall'alimentazione.

Passo 2

Collegate il circuito come illustrato nello schema precedente.

Passo 3

Guardate le denominazioni ABCD degli Outboard LR-2 e LR-6 mostrate nello schema. La nomenclatura dei quattro terminali di ingresso o uscita degli Outboard è arbitraria. Riuscite a capire perché è così? Spiegatelo nello spazio qui sotto.

Sugli Outboard LR-2 ed LR-6 di solito non c'è la nomenclatura e quindi potete chiamare i terminali come meglio vi piace, ABCD, DCBA, 0123, 3210, 1248, 8421 o in qualsiasi altro modo purché sia compatibile. Qui sotto diamo alcuni esempi di nomenclature.

Quando abbiamo detto che la nomenclatura dei due Outboard è arbitraria, intendevamo dire che: *o voi o la natura stessa dell'esperimento determinano il tipo di nomenclatura.*

Passo 4

Collegate i +5 V all'alimentazione. Fate scattare i quattro switch logici in modo che siano tutti in ON, cioè nella posizione corrispondente allo stato logico 1. I quattro indicatori a LED sono accesi?

Nel nostro caso sono tutti e quattro accesi.

Passo 5

Fate scattare i quattro switch in modo che siano tutti in OFF, cioè nella posizione corrispondente allo stato logico 0. I 4 LED sono accesi o spenti?

Nel nostro caso sono tutti spenti e quindi possiamo concludere che:

Uno switch logico in OFF, cioè allo stato logico 0 corrisponde ad un LED spento, cioè allo stato logico 0.

Uno switch logico in ON, cioè allo stato logico 1 corrisponde ad un LED acceso, cioè allo stato logico 1.

Passo 6

Cambiate le posizioni dei quattro switch logici passando da ON a OFF e viceversa e scrivete, nella tavola della pagina seguente, lo stato logico dei quattro indicatori a LED che osservate durante l'esperimento. Ricordate che 1=switch logico in ON = LED acceso e che 0 = switch logico in OFF = LED spento.

Tabella 1. Risultati del Passo 6 dell'Esperimento N. 3.

Stato logico degli switch	Stato logico osservato sui LED
DCBA	DCBA
0 0 0 0	0 0 0 0 (Passo 5)
0 0 0 1	
0 0 1 0	
0 0 1 1	
0 1 0 0	
0 1 0 1	
0 1 1 0	
0 1 1 1	
1 0 0 0	
1 0 0 1	
1 0 1 0	
1 0 1 1	
1 1 0 0	
1 1 0 1	
1 1 1 0	
1 1 1 1	1 1 1 1 (Passo 4)

Potete osservare che esiste una stretta corrispondenza tra gli stati logici degli switch e gli stati logici osservabili sugli indicatori a LED.

Passo 7

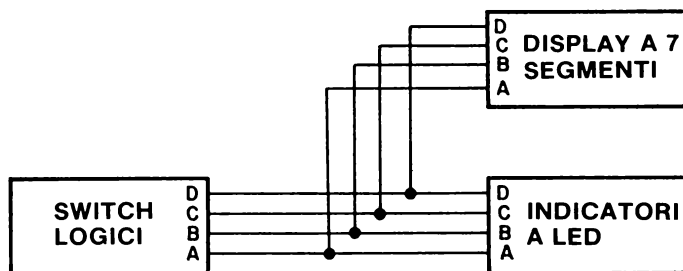
Perché c'è questa stretta corrispondenza?

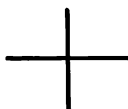
Perché i due dispositivi sono collegati tra loro, A con A, B con B, C con C e D con D.

ESPERIMENTO N. 5

Scopo

Lo scopo di questo esperimento è di dimostrare come funziona l'LR-4 seven segment LED display Outboard.

Schema del circuito

Passo 1

Studiate bene lo schema di questo circuito. Osservate che quando due fili si incrociano ma non sono collegati, usiamo il seguente simbolo

Invece per indicare che due fili sono collegati tra loro, usiamo le forme seguenti:

se non si incrociano e sono collegati

se si incrociano e sono collegati

In tutto il libro useremo questa convenzione.

Passo 2

Staccate i +5V dall'alimentazione e collegate il circuito descritto nello schema.

Oltre ai ponticelli, bisogna collegare ai circuiti integrati solo otto fili. Quando diciamo "collegate il circuito illustrato nello schema", intendiamo dire:

1. Togliere tutti i collegamenti e i componenti elettronici digitali non necessari dalla piastra di montaggio.

2. Inserite gli Outboard che vi servono sulla piastra di montaggio.

3. Eseguite i collegamenti illustrati nello schema del circuito.

4. Controllate, dopo averlo completato, tutto il circuito digitale per essere sicuri che sia esatto.

5. Ricollegate l'alimentazione alla piastra collegando il bus dei terminali di alimentazione alla sorgente di tensione.

Passo 3

Fate scattare i quattro switch logici in modo che siano tutti in OFF, cioè allo stato logico 0; a questo punto dovreste vedere uno "0" sull'Outboard LR-4.

Fate scattare i quattro switch logici in modo che siano tutti in ON, cioè allo stato logico 1; adesso dovreste osservare che il display a sette segmenti presenta tutti blank, come è giusto che sia.

Fate scattare i quattro switch logici in modo che solo A sia 1, e che B, C e D siano 0; adesso dovrete osservare un "1" sul display a sette segmenti. Potrebbe darsi che compaia un "8"; se è così probabilmente avete scambiato lo switch D con l'A. Ora fate scattare gli switch in modo che solo C sia 1; dovreste vedere un "4" sul display.

Passo 5

Variate la posizione degli switch logici da 0 a 1 e viceversa e scrivete nella tabella della pagina seguente, i numeri decimali o i simbolo che compaiono sul display a sette segmenti. Vedrete dieci numeri, un blank e cinque simboli diversi.

Tabella 2. Risultati del passo 5 dell'Esperimento N. 4.

Stato logico degli switch DCBA	Numero o simbolo osservato sul display a sette segmenti
0 0 0 0	0
0 0 0 1	1
0 0 1 0	2
0 0 1 1	
0 1 0 0	4
0 1 0 1	
0 1 1 0	
0 1 1 1	
1 0 0 0	8
1 0 0 1	
1 0 1 0	
1 0 1 1	
1 1 0 0	
1 1 0 1	
1 1 1 0	
1 1 1 1	BLANK

Passo 6

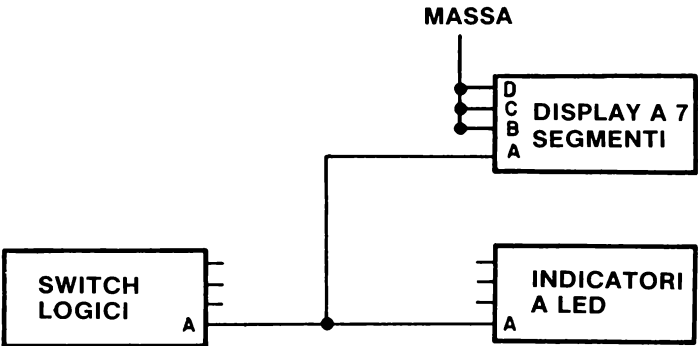
Le prime dieci entrate (o elementi) della tabella 2 sono un esempio di binari *decimali codificati*, o BCD (Binary Coded Decimal). Cosa significa, secondo voi?

Il sistema BCD, è un sistema di rappresentazione di numeri, in cui ogni digit decimale o numero è espresso da un numero binario. Nella tabella 2 ci sono solo numeri di un digit e quindi sono sufficienti quattro bit per codificarli. Per esempio, il numero binario DCBA=0100 rappresenta il 4 decimale.

ESPERIMENTO N. 6

Scopo

Lo scopo di questo esperimento è di mostrare come funziona l'Outboard LR-4 usato come display a LED zero/uno.

Schema del circuito

Passo 1

Costruite sulla vostra piastra di montaggio il circuito mostrato qui sopra. Usate gli Outboards LR-2, LR-4 e LR-6. Date alimentazione al circuito.

Passo 2

Nello spazio qui sotto scrivete quello che potete osservare quando fate passare lo switch logico A dallo stato logico 0 allo stato logico 1.

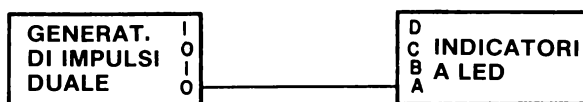
Switch logico A	Indicatore a LED A	Display a sette segmenti
0	0	
1	1	

Se avete fatto questo esperimento senza commettere errori, dovreste vedere sul display a sette segmenti, "0" quando lo switch logico A è allo stato logico 0 e "1" quando lo switch logico A è allo stato logico 1. Il display a sette segmenti, collegato in questo modo, si comporta come un semplice indicatore a LED col solo vantaggio che potete vedere direttamente "0" o "1", mentre con l'indicatore a LED dovete sapere che se è acceso corrisponde ad 1 e se è spento corrisponde a 0.

ESPERIMENTO N. 7

Scopo

Lo scopo di questo esperimento è di dimostrare come funziona l'LR-7 dual pulser Outboard.

Schema del circuito

Passo 1

Togliete dalla piastra di montaggio gli Outboard ed i fili che non vi servono e collegate il circuito mostrato nello schema.

Passo 2

Per far funzionare il generatore di impulsi è sufficiente che facciate una leggera pressione su di esso e che poi la togliate. Fatelo più volte e scrivete nello spazio qui sotto quello che riuscite ad osservare.

Generatore d'impulsi	Indicatori a LED
Premuto	
Rilasciato	

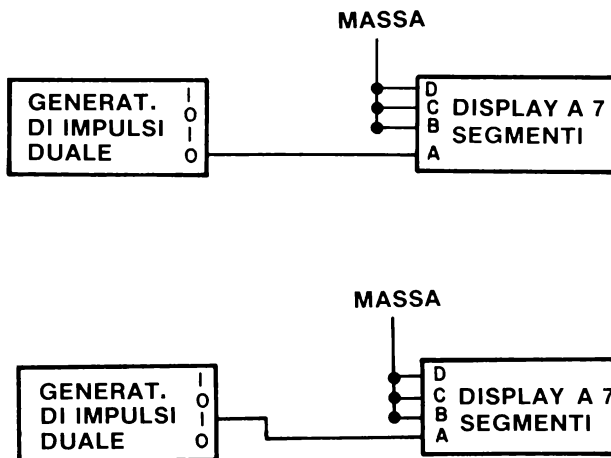
Un *impulso di clock* viene definito come un ciclo logico completo, cioè un passaggio dallo stato logico 0 allo stato logico 1 e poi ancora a 0 (impulso di clock *positivo*), oppure un passaggio da 1 a 0 e poi ancora ad 1 (impulso di clock *negativo*). Spiegate come fareste a generare un impulso di clock positivo con il generatore di impulso appena fatto.

Basta premere sul generatore di impulsi e poi rilasciarlo. In un capitolo successivo imparerete che questo generatore di impulsi, e solo lui, genera un impulso di clock ogni volta che lo premete e lo rilasciate. Questo succede perchè il generatore di impulsi possiede una logica antirimbalzo.

ESPERIMENTO N. 8

Scopo

Lo scopo di questo esperimento è di mostrare il comportamento delle uscite "0" e "1" dell'Outboard LR-7.

Schema del circuito

Passo 1

Collegate sulla piastra di montaggio il primo dei due circuiti, azionate il generatore di impulsi e rilasciatelo. Adesso cambiate il collegamento tra il generatore di impulsi ed il display come illustrato nel secondo circuito; azionate ancora il generatore di impulsi.

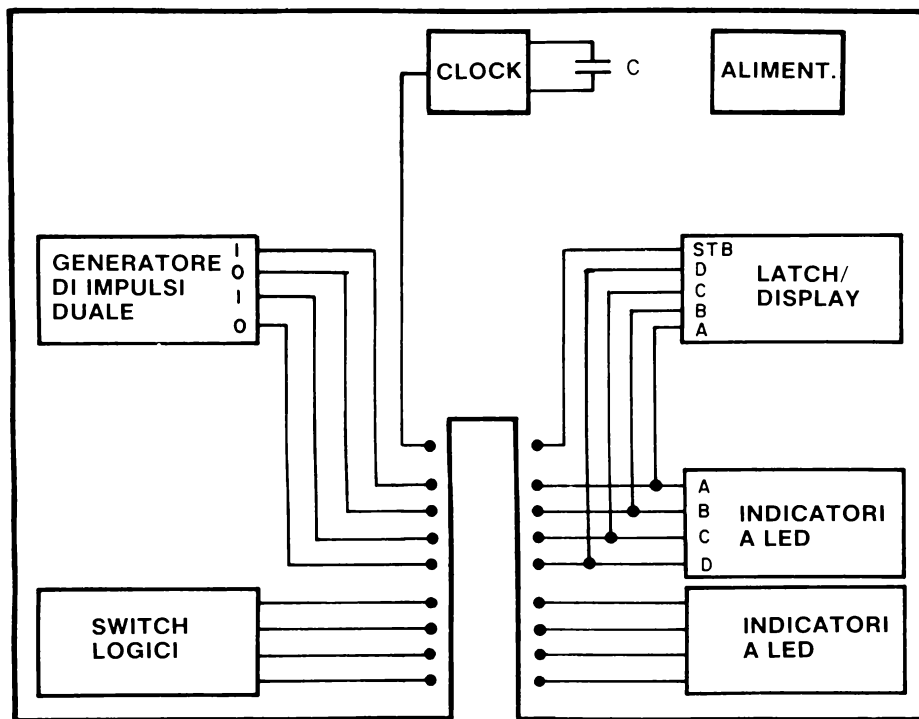
Quale differenza c'è nel comportamento dei due circuiti?

Nel primo circuito, il display passa ad "1" quando il generatore d'impulsi è premuto e ritorna a "0" quando è rilasciato. Nel secondo circuito succede esattamente il contrario, cioè, quando il generatore d'impulsi è premuto il display passa a "0" e ritorna a "1" quando è rilasciato. Questo esperimento vi ha quindi dimostrato come potete generare un impulso di clock sia positivo che negativo.

ESPERIMENTO N. 9

Scopo

Lo scopo di questo esperimento è analizzare il modo di funzionamento dell'LR-25 Breadbording station Outboard.

Schema del circuito

Passo 1

Studiate attentamente lo schema dell'Outboard LR-25. Osservate che contiene due gruppi di quattro indicatori a LED, un gruppo di quattro switch, due generatori di impulsi con logica antirimbalzo, un clock ed un diodo-protezione ingresso di alimentazione.

Inoltre contiene lo zoccolo per un display numerico o esadecimale Hewlett-Packard.
Vi mostriamo qui una fotografia dell'Outboard

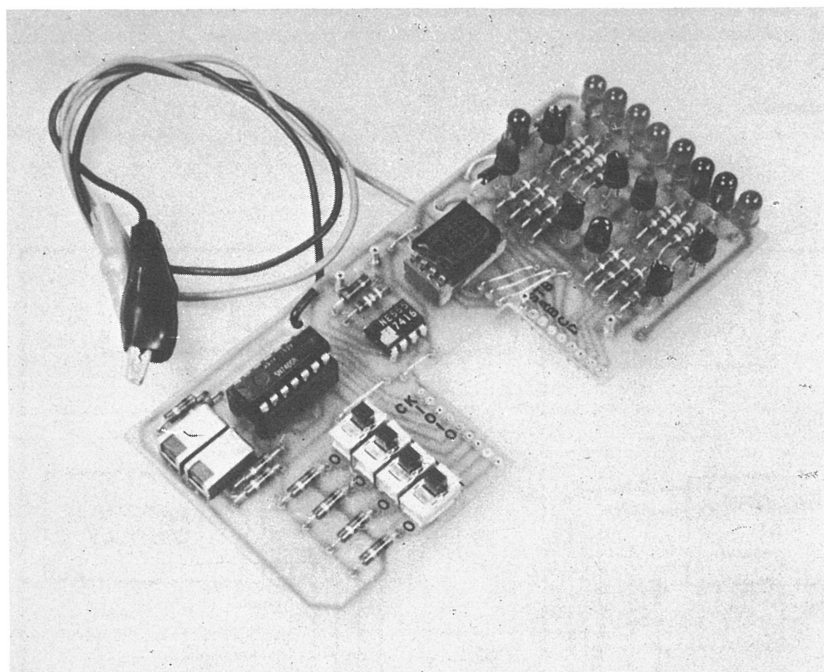



Figura 9-11. L'Outboard breadbording station. Confrontate questa fotografia con lo schema dell'Outboard nella figura precedente.

Passo 2

Inserite l'Outboard breadboarding station ad una estremità della piastra SK-10 o SK-50. Nella figura 9-12 è mostrato l'Outboard inserito in una piastra SK-50. Accertatevi che i due pin di alimentazione, +5V e massa, siano ben collegati ai bus terminali di alimentazione.

Alimentate la piastra di montaggio collegando le prese a coccodrillo ad una sorgente di alimentazione. Il LED dell'ingresso di alimentazione, dovrebbe accendersi. Per controllare il funzionamento di questo Outboard, dovete ripetere l'esperimento precedente.

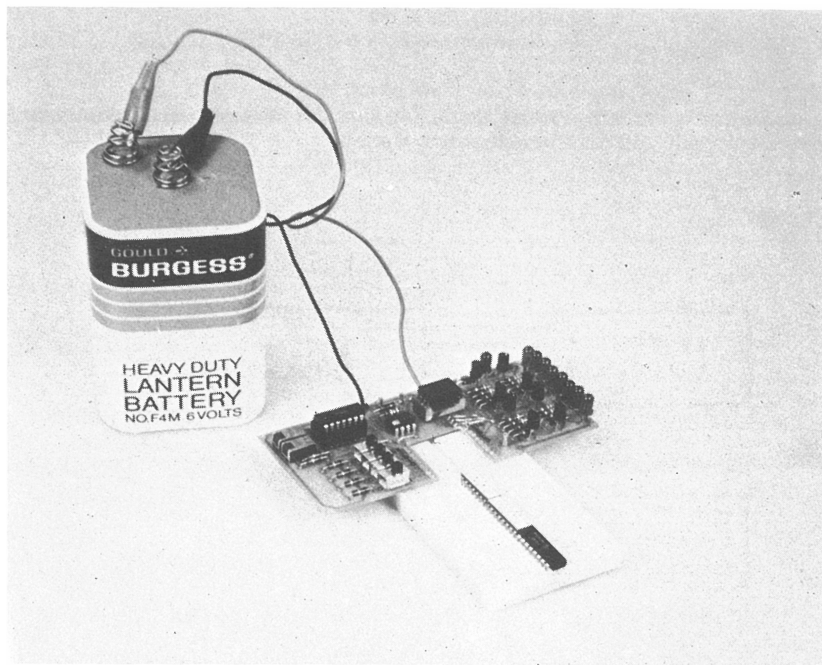



Figura 9-12. Un esempio di come l'Outboard breadbording station LR-25 viene inserito su una piastra di montaggio SK-10 o SK-50. In questo caso la piastra è un SK-50.

Passo 3

Eseguite l'Esperimento N. 1 di questo capitolo usando i LED dell'Outboard LR-25.

Passo 4

Eseguite l'Esperimento N. 2 di questo capitolo usando i LED dell'Outboard LR-25.

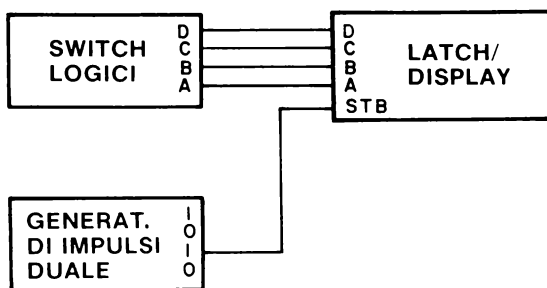
Passo 5

Eseguite l'Esperimento N. 3 di questo capitolo usando i LED e gli switch dell'Outboard LR-25.

ESPERIMENTO N. 10

Scopo

Lo scopo di questo esperimento è di mostrare come funziona l'LR-26 latch/display Outboard o la funzione equivalente dell'Outboard breadbording station.

Schema del circuito

Passo 1

Per fare questo esperimento avete bisogno solo dell'Outboard breadbording station LR-25 oppure degli Outboard LR-2, LR-7 e LR-26. Collegate il circuito mostrato nello schema e date alimentazione alla piastra.

Passo 2

Il collegamento più importante è quello che va dall'uscita "1" del generatore di impulsi all'ingresso STB del latch/display; la ragione è che il display contiene al suo interno una memoria, il latch, la quale memorizza il dato solo quando l'ingresso STB è allo stato logico 0. Se l'ingresso STB è 1, il latch è *disabilitato* e non si produrrà alcun cambiamento nel display, qualsiasi sia lo stato logico degli switch.

Mettete tutti gli switch a 0, poi premete e rilasciate il generatore d'impulsi; vedrete sul latch/display il numero "0".

Mettete tutti gli switch a 1, poi premete e rilasciate il generatore d'impulsi; adesso il display mostrerà una configurazione di blank.

Passo 3

Cambiate lo stato dei quattro switch tra gli stati logici 0 e 1, premendo e rilasciando il generatore d'impulsi dopo ogni cambiamento e scrivete, nella tabella 3 i numeri decimali o i simboli che vedete.

Tabella 3. Risultati del Passo 3 dell'Esperimento N. 10.

Stato logico degli switch	Numero, simbolo o lettera osservata sul latch/display
DCBA	
0 0 0 0	0
0 0 0 1	
0 0 1 0	
0 0 1 1	
0 1 0 0	
0 1 0 1	
0 1 1 0	
0 1 1 1	
1 0 0 0	
1 0 0 1	
1 0 1 0	
1 0 1 1	
1 1 0 0	
1 1 0 1	
1 1 1 0	
1 1 1 1	BLANK

Passo 4

Se avete un latch/display decimale, osserverete i numeri da 0 a 9, uno stato in cui tutti i punti sono accesi, un meno, e quattro blank. Se invece avete un latch/display esadecimale, vedrete i numeri da 0 a 9 e le lettere da A a F. Che tipo di display avete?

Noi abbiamo un latch/display decimale che abbiamo comprato perché aveva un prezzo molto inferiore a quello del latch/display esadecimale. Esadecimale significa sedici e quindi un display esadecimale è un display che ha sedici stati differenti.

Passo 5

Cosa succede se collegate il filo dell'ingresso STB all'uscita "0" del generatore d'impulsi?

Il latch/display sarà permanentemente *disabilitato* e quindi non sarà in grado di percepire immediatamente il cambiamento di stato degli switch, ma occorrerà che premiate e rilasciate il generatore d'impulsi perché possa stabilire il cambiamento avvenuto negli switch.

Fate questo cambiamento e dimostrate da soli che il latch/display è proprio disabilitato.

DOMANDE RIEPILOGATIVE

Le seguenti domande come ausilio per ripassare le caratteristiche operative degli Outboard che avete studiato in questo capitolo.

1. Cosa succederebbe, secondo voi, se applicate +12V invece che +5V al bus di terminali più esterno della piastra? Supponete di avere un circuito digitale già collegato sulla piastra.

1. Cosa succederebbe, secondo voi, se applicate −12V invece che il potenziale di massa al bus di terminali più interno della piastra? Supponete, anche qui, di avere un circuito digitale già collegato sulla piastra.

3. Che differenza c'è tra un generatore d'impulsi e uno switch logico? Avete fatto in questo capitolo qualche esperimento che ha chiaramente mostrato questa differenza?

4. Se i pin di ingresso A,B, e C di un display a LED a sette segmenti solo allo stato logico 0 e il pin D è allo stato logico 1 che uscita vedete sul display?

5. Scrivete il complemento di ognuna delle parole che compaiono nella seguente tabella.

Parola	Complemento della parola
ON	
NON VA	
NERO	
OFF	
START	
STOP	

6. Qual'è la funzione base dei seguenti Outboard?

    a. Outboard LR-2

    b. Outboard LR-4

    c. Outboard LR-6

    d. Outboard LR-7

RISPOSTE

1. Molto probabilmente brucereste tutti i circuiti integrati che sono sulla piastra, compresi i chip dell'LR-4.

2. Molto probabilmente brucereste tutti i circuiti integrati che ci sono.

3. Un generatore d'impulso può generare un singolo impulso di clock, positivo o negativo, mentre uno switch logico non può perché non ha logica antirimbalzo. In questo capitolo non è stato fatto alcun esperimento che chiarisce questa diversità di comportamento.

4. Il decimale "8".

5.

Parola	Complemento della parola
ON	OFF
NON VA	VA
NERO	BIANCO
OFF	ON
START	STOP
STOP	START

6. a. Fornisce quattro switch, ognuno dei quali serve come variabile logica 0 o 1 di ingresso ad un circuito digitale.

   b. Fornisce un display a quattro bit che può essere usato per un'uscita da un circuito digitale. Usato normalmente come display a dieci stati, o display decimale, può essere usato anche come display a sedici stati, o esadecimale, poiché esistono sedici stati riproducibili dell'uscita.

   c. Fornisce quattro indicatori a LED individuali ognuno dei quali serve come display a un bit per l'uscita di un circuito digitale. I quattro indicatori a LED, se vengono usati insieme, possono servire come display di un'uscita di quattro bit.

   d. Fornisce due sorgenti indipendenti di un singolo impulso di clock debounced la grandezza del quale non è minore di 200 ms. Sia l'uscita normale che quella complementare sono disponibili ad ogni generatore d'impulsi.

CAPITOLO 10

CIRCUITI INTEGRATI

INTRODUZIONE

Questo capitolo vi introdurrà alle caratteristiche della famiglia di circuiti integrati 7400; vi sarà descritto un certo numero di chip tra i più semplici e vi saranno dati consigli e suggerimenti sull'utilizzo di questi IC (circuiti integrati). Otto esperimenti vi permetteranno di acquistare una certa familiarità con i più importanti chip della famiglia 7400.

OBIETTIVI

Alla fine di questo capitolo sarete in grado di:

- Dare la definizione di circuito integrato.
- Descrivere la configurazione dei pin dei più comuni chip.
- Indicare qual'è lo stato logico di un ingresso non collegato di un circuito integrato della famiglia 7400.
- Riconoscere un circuito integrato della serie 7400 dal suo numero di identificazione.
- Individuare il pin 1 di un circuito integrato della serie 7400.
- Conoscere la differenza tra fan-in e fan-out.
- Riassumere le caratteristiche più importanti della sottofamiglia dei circuiti integrati TTL.
- Saper spiegare come inserire un circuito integrato sul breadboard.
- Collegare sul breadboard e successivamente determinare la tabella della verità di un semplice chip.

COSA È UN CIRCUITO INTEGRATO?

Il *circuito integrato* è stato definito da Graf nel seguente modo:

Circuito integrato — Abbreviazione = IC. (1) Una combinazione di elementi circuitali interconnessi, inseparabilmente collegati su o dentro un substrato ininterrotto. (2) Qualsiasi dispositivo elettronico nel quale gli elementi, sia attivi che passivi, sono contenuti in un singolo blocco fisico. Nell'elettronica digitale, il termine è applicato soprattutto ai circuiti contenenti elementi semiconduttori.

In altre parole, un circuito integrato è un dispositivo elettronico, di solito costruito di materiale semiconduttore, nel quale transistori, resistori, diodi, vengono contemporaneamente costruiti per formare un piccolissimo circuito elettronico con una funzione definita. Il blocco del circuito integrato che contiene i transistori, i resistori, ecc. può essere non più grande di 30 mils per lato.

Per apprezzare un circuito integrato, bisogna conoscere lo sviluppo che ha avuto l'industria elettronica negli ultimi trent'anni. Trent'anni fa, le apparecchiature elettroniche erano realizzate con *componenti discreti*, cioè componenti che venivano fabbricati prima di essere inseriti nell'apparecchiatura. Tra questi componenti, i più comuni erano resistori, condensatori, induttanze e le diffusissime valvole. Le apparecchiature erano ingombranti e rigide, e avevano un elevato consumo. Nei sistemi piu grandi, come i computer a valvole, occorreva lottare con le valvole che si bruciavano con una frequenza impressionante.

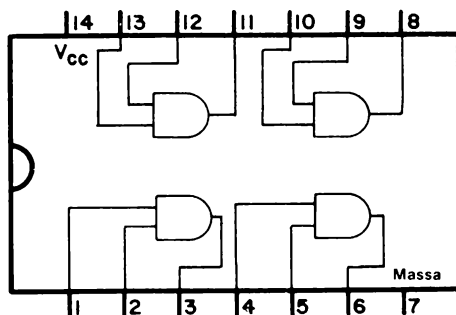
Voi saprete che i transistor furono scoperti nel 1948 e che questo dispositivo entrò subito in concorrenza con le valvole, sostituendole completamente nel 1970. Ma anche i primi transistor erano dei componenti discreti. Con lo sviluppo della tecnologia avutasi negli anni '50 i transistor divennero resistenti anche alle alte correnti ed alle alte frequenze e questa possibilità di operare alle alte frequenze divenne importantissima nella seconda generazione dei computer che erano basati su componenti allo stato solido discreti.

L'idea base dei circuiti integrati fu formulata tra la fine degli anni '50 e l'inizio degli anni '60 e già nel 1965 i sistemi di fabbricazione dei semiconduttori si erano sviluppati in modo tale da permettere di costruire circuiti che contenessero, in un singolo wafer di silicio, anche trenta transistori. Questi non erano più dei circuiti discreti. Resistori, transistor e diodi venivano fabbricati a cinquecento, mille per volta. I primi blocchi, o wafer, del resto ancora abbastanza grandi, furono poi ridotti in piccoli "chip" che venivano realizzati in contenitori plastici o ceramici che, a loro volta, potevano essere saldati su piastre a circuiti stampati.

Tra la fine degli anni '60 e l'inizio degli anni '70 si passò dalla *piccola scala di integrazione* (SSI), alla *media* (MSI) e poi alla *larga scala d'integrazione* (LSI). La piccola scala d'integrazione si riferisce a quei circuiti che contengono solo porte semplici, buffer e flip-flop; i chip di questo tipo non contengono più di dieci o venti porte. Nella media scala d'integrazione abbiamo sistemi logici semplici e completi quali contatori, piccole memorie ad accesso random, decodificatori, multiplexer e registri di shift; di solito contengono da venti a cento porte. La larga scala di integrazione si riferisce invece ai circuiti integrati con una complessità superiore a cento porte. Oggi i normali microprocessor contengono l'equivalente di alcune migliaia di porte in un singolo wafer di silicio.

Non si può sottovalutare il grande cambiamento verificatosi nella fabbricazione dei dispositivi elettronici a semiconduttore. Il costo di fabbricazione di un transistor in un circuito integrato si è molto abbassato e la qualità tecnica di ogni transistor nei circuiti integrati è migliorata. Le porte in uso oggi, fatte coi transistor, hanno bisogno di meno tensione e sono più veloci. Minore è la tensione richiesta da ogni singola porta, e più porte si possono mettere su un singolo wafer. I microcomputer di oggi sono molto simili, come prestazioni, ai più grossi computer degli anni '50.

RAPPRESENTAZIONE SIMBOLICA DEI CIRCUITI INTEGRATI

In questi capitoli vi chiederemo di realizzare un certo numero di circuiti. Sorge allora spontaneo domandarsi: come rappresentare schematicamente i circuiti integrati in modo da facilitare il lavoro di assemblaggio? Lasciateci rispondere a questa domanda prendendo in considerazione i diversi modi con cui è possibile rappresentare i collegamenti di un chip *7408, QUAD 2-input positive AND gate,* la cui configurazione è mostrata di seguito:

Il numero 7408 viene dato a tutti i chip che hanno la stessa configurazione di pin e le stesse specifiche elettriche del chip 7408 originale, costruito dalla Texas Instruments Incorporated alla fine del 1960. Oggi questo chip viene costruito da almeno dieci diverse industrie, Signetics, National Semiconductor, Sprague, Stewart Warner Corporation, Texas Instruments Incorporated, SGS - Ates, Siemens, AEG, Motorola, Harris Semiconductor, Fairchild Semiconductor e Raytheon Semiconductor. Il termine *quad* significa che in ogni chip ci sono quattro diversi 2-input positive AND gate. L'aggettivo *positive* si riferisce al fatto che le porte funzionano come porte AND quando è usata una logica positiva (per il momento non parliamo della logica negativa). Infine è chiaro, siccome ogni porta AND ha due input, o ingressi, il significato del termine *2-input.*

Ogni chip della serie di circuiti integrati 7400 ha solo due ingressi di alimentazione, MASSA (GND) e Vcc. Il simbolo Vcc, rappresenta i +5V; perchè il chip funzioni basta che la Vcc sia compresa tra +4,75V e +5,25V, però se applicate più di +6V rischiate di bruciare tutto il chip. Anche se invertite i collegamenti di alimentazione e applicate +5V all'ingresso di MASSA, e applicate la MASSA all'ingresso Vcc, rischiate di bruciare il chip.

Alcuni resistono da trenta secondi a un minuto prima di bruciare e quindi in alcuni casi si riesce a salvarli. Di solito un chip quando brucia si scalda molto.

Nella figura 10-1, sono illustrati quattro chip che contengono rispettivamente quattordici, sedici, ventiquattro e quaranta pin. Il chip più a sinistra è un 7400, 2-input positive NAND gate, molto simile al 7408 di cui abbiamo già parlato. Adesso vorremmo parlare di come sia possibile identificare il pin 1 di un chip.

Ciascuno dei due chip a sinistra ha una tacca semicircolare nella parte superiore, questa tacca segna il lato del chip dove si trovano i pin col numero più basso e più alto rispettivamente. Nella serie 7400 questi pin sono il numero 1 e 14 e si trovano su i due lati opposti della porta del chip in cui si trova la tacca, proprio come nella configurazione dei pin del 7408 che vi abbiamo già mostrato. Il pin 1 si trova a *sinistra della tacca* guardando il chip dal di sopra e dal lato più lontano dalla tacca. Dopo aver stabilito qual'è il pin 1, si numerano gli altri girando in senso anti-orario lungo il chip, partendo da un lato e proseguendo sugli altri. Se guardate la configurazione dei pin del 7408 riuscite a capire meglio come si fa. Questa procedura è valida indipendentemente da quanti pin ci sono sul chip. Per un chip di 40 pin, i pin 1 e 40 stanno dalla stessa parte del chip, e i pin 20 e 21 stanno dall'altra. Il chip bianco sulla destra della Figura 10-1, possiede 40 pin; il pin 1 viene identificato per mezzo del piccolo punto nero che si trova nell'angolo del chip in basso a destra.

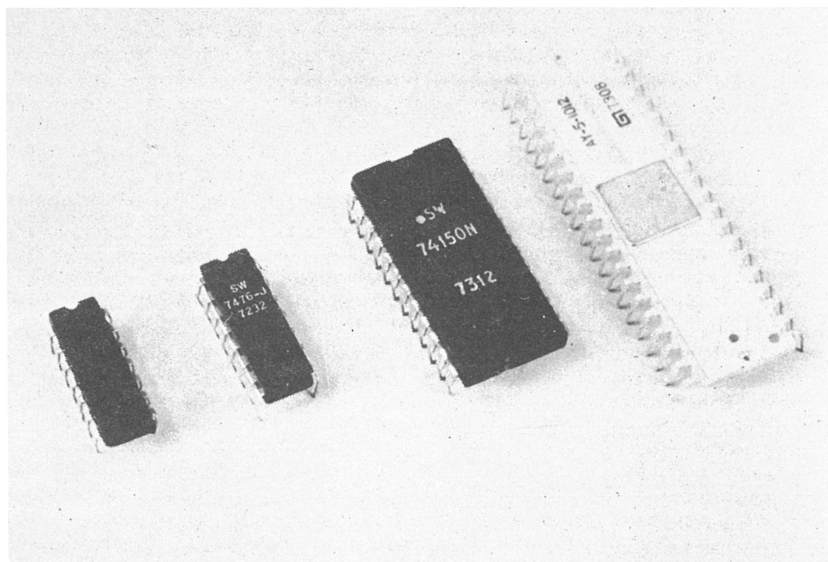
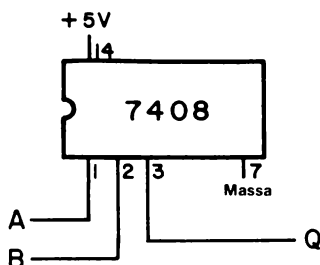
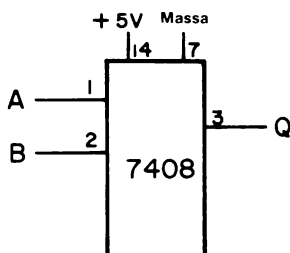
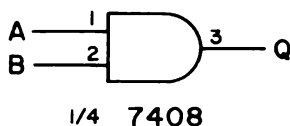
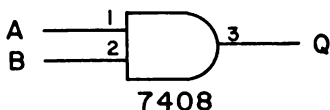



Figura 10-1. La fotografia mostra quattro chip, ognuno rappresentativo di una delle più usate configurazioni di pin: 14 pin, 24 pin e 40 pin. Si nota senza difficoltà il numero identificativo del chip.

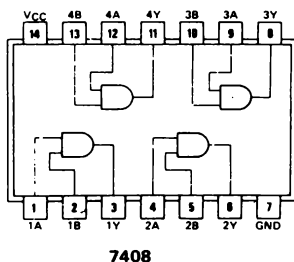
Supponiamo che vogliate collegare un singolo 2-input positive AND gate, nel quale A e B sono gli ingressi e Q sia l'uscita. Per aiutarvi in questo lavoro, vi forniamo uno schema nel quale abbiamo tracciato la riproduzione in scala del chip 7408 e dove indichiamo il numero dei pin ed i collegamenti che dovete fare,

Quindi, come mostrato in questo disegno, collegherete il pin 14 ai +5V e il pin 7 alla MASSA; gli ingressi saranno applicati ai pin 1 e 2 e l'uscita sarà al pin 3. *Di solito però non si usa questo metodo* per il motivo che, quando il circuito è più complicato perchè ci sono più chip, lo schema diventa troppo confuso.

Un altro metodo è il seguente:

Secondo questo metodo i pin non vengono più messi in sequenza, ma il chip viene diviso *in blocchi logici*, per cui i due pin di ingresso sono messi a sinistra, l'uscita a destra e i collegamenti di alimentazione in alto. *Questo schema non comprende la tacca semicircolare perchè non rappresenta la reale configurazione dei pin del circuito integrato 7408.* In questa trattazione useremo sempre questo tipo di *schema a blocchi,* con la sola eccezione delle porte.

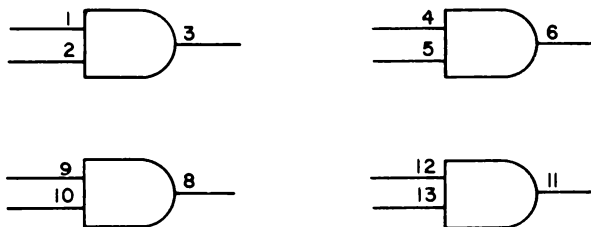
Un terzo modo di rappresentare le porte è mostrato di seguito:

1/4 7408

Osservate che abbiamo usato il vero simbolo della porta AND a 2-input e che vi abbiamo aggiunto il numero dei pin degli ingressi A e B e dell'uscita Q. La notazione "1/4 7408" significa che vi sono quattro porte AND a 2 ingressi nel chip 7408 e che noi ne abbiamo usato 1/4, cioè una sola porta. Adesso facciamo un piccolo cambiamento a questo metodo che è molto usato per rappresentare le porte semplici e la loro configurazione di pin: trascureremo i "1/4", "1/3", "1/2" o "1/6" che precedono il numero dei chip, e quindi, in definitiva, la rappresentazione di una semplice porta AND a 2 ingressi diventa:

Per eseguire correttamente questi collegamenti, dovete accertarvi che le connessioni di alimentazione siano state fatte, senza errori, ai pin 7 e 14 del circuito integrato 7408. Se non date tensione al chip, la porta non funziona, infatti ogni circuito integrato ha bisogno di essere alimentato.

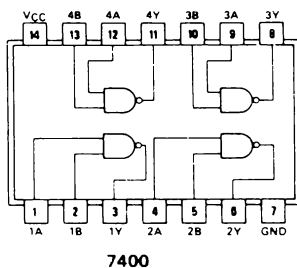
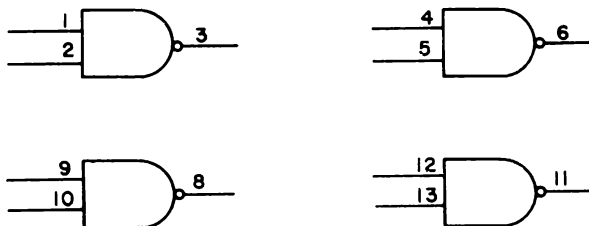
7408, PORTA AND A DUE INGRESSI (TWO INPUT AND GATE)

La *configurazione dei pin*, o *assegnazione dei pin*, di un chip 7408, come descritta nel "The TTL Data Book for Design Engineers" della Texas Instruments Incorporated, è la seguente:

Osservate gli ingressi di alimentazione, MASSA (GND) e Vcc, e la stampigliatura di ognuna delle quattro diverse porte AND a 2-ingressi. Questo chip contiene quattro porte AND a 2 ingressi, *indipendenti*. Se si brucia una porta, si possono ancora usare le altre tre. Nello schema di configurazione dei pin che vi abbiamo appena fatto vedere, le quattro porte vengono identificate nel seguente modo:

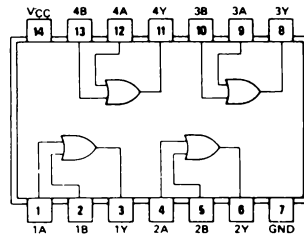
Prima porta	: Ingresso 1A e 1B,	Uscita 1Y
Seconda porta	: Ingresso 2A e 2B,	Uscita 2Y
Terza porta	: Ingresso 2A e 3B,	Uscita 3Y
Quarta porta	: Ingresso 4A e 4B,	Uscita 4Y

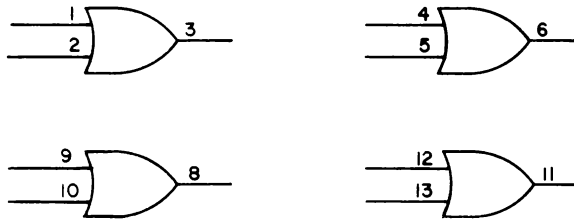
Gli schemi di queste quattro porte sono indicati qui di seguito:

Occorre dare tensione al 7408 prima che una qualsiasi delle quattro porte cominci a funzionare.

7400 PORTA NAND A DUE INGRESSI (TWO INPUT NAND GATE)

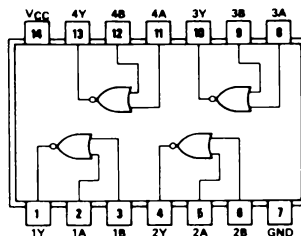
La configurazione dei pin del chip 7400, porta NAND a 2 ingressi, è in apparenza molto simile a quella del circuito integrato 7408.

I pin di alimentazione sono l'1 e il 14, come per il 7408 e il chip contiene quattro porte NAND a 2 ingressi *indipendenti*, la cui rappresentazione è la seguente:

Bisogna fornire tensione al chip 7400, prima che una di queste porte inizi a funzionare.

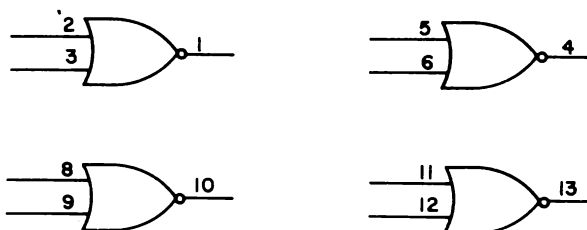
7432, PORTA OR A DUE INGRESSI (TWO INPUT OR GATE)

Anche la configurazione dei pin del 7432, QUAD 2 input positive OR gate

7432

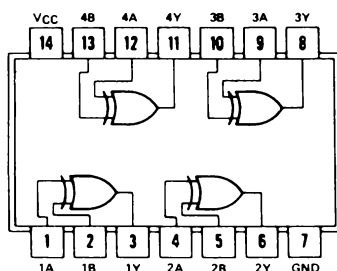
ricorda quella dei chip 7400 e 7408. Osservate infatti che in tutti questi tre chip gli ingressi e le uscite corrispondono. Nel chip 7432 ci sono quattro porte *indipendenti,*

Bisogna fornire tensione al chip 7432 prima che una di queste porte inizi a funzionare.

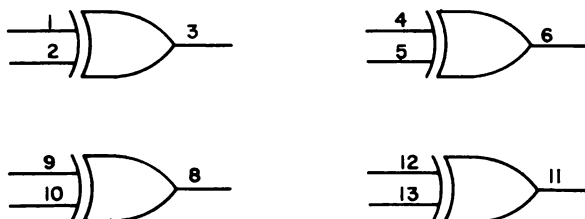
7402, PORTA NOR A DUE INGRESSI (TWO INPUT NOR GATE)

La configurazione dei pin del 7432, QUAD 2-input positive NOR gate

7402

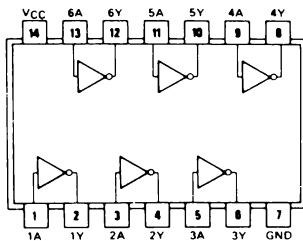
è diversa da quelle che abbiamo visto fino ad ora; gli ingressi e le uscite sono diverse; ricordatevelo quando usate questo chip. Le quattro porte NOR *indipendenti*, possono essere rappresentate nel modo seguente,

Bisogna dare tensione al chip 7432, prima che qualcuna delle porte cominci a funzionare.

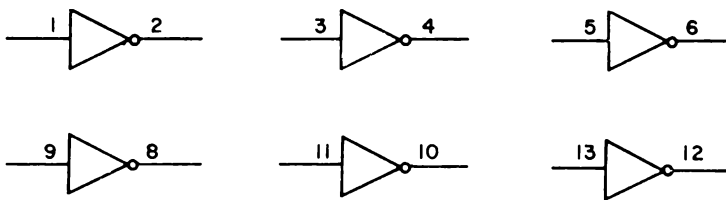
7486, PORTA OR ESCLUSIVO (EXCLUSIVE - OR GATE)

Il 7486, quadrupla porta OR-Esclusivo

7486

contiene quattro porte OR-Esclusico *indipendenti* e può essere rappresentato nel seguente modo:

Bisogna dare tensione al 7486 prima che una porta OR-Esclusivo cominci a funzionare.

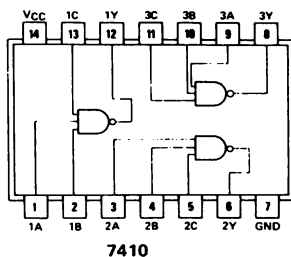
7404, INVERTITORE (INVERTER)

Il chip 7404, invertitore

7404

contiene sei invertitori indipendenti ed è un chip molto diffuso perchè di invertitori ce n'è sempre bisogno. La rappresentazione dei sei invertitori è la seguente:

Bisogna dare tensione al 7404, prima che uno degli invertitori cominci a funzionare.

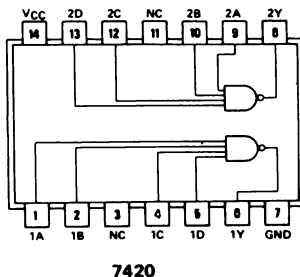
7410, PORTA NAND A TRE INGRESSI (THREE INPUT NAND GATE)

Il chip 7410, porta NAND positiva a 3 ingressi,

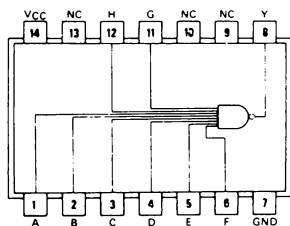
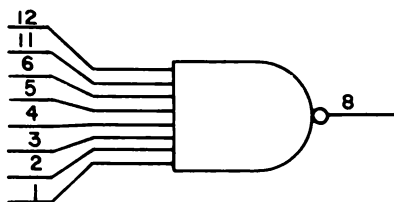
7410

contiene tre porte *indipendenti* che possono venir rappresentate nel modo seguente,

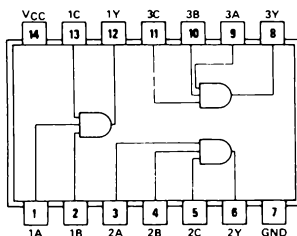
7420, PORTA NAND A QUATTRO INGRESSI (FOUR INPUT NAND GATE)

Il chip 7420, doppia porta NAND a 4 ingressi

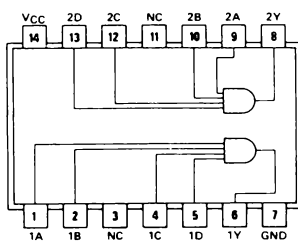
7420

contiene solo due porte *indipendenti* che possono venir rappresentate nel modo seguente:

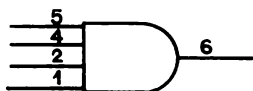
7430, PORTA NAND A OTTO INGRESSI (EIGHT INPUT NAND GATE)

Il chip 7430, porta NAND positiva a 8 ingressi, contiene una singola porta NAND e può essere rappresentato nel modo seguente:

7411, PORTA AND A TRE INGRESSI (THREE INPUT AND GATE)

Il chip illustrato nello schema è il 74H11, porta NAND positiva a tre ingressi,

La H significa "alta velocità (high speed) TTL", di cui parleremo più tardi, sempre in questo capitolo. Le tre porte AND *indipendenti* possono essere rappresentate nel seguente modo:

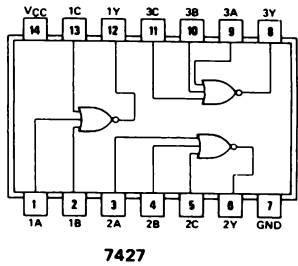
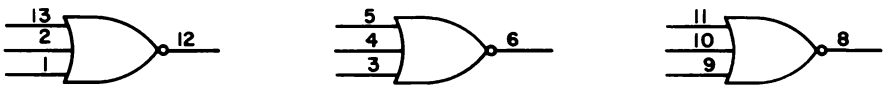
7421, PORTA AND A QUATTRO INGRESSI (FOUR INPUT AND GATE)

Il chip 7421, doppia porta AND a 4 ingressi, assomiglia al 7420, sia nella configuarzione dei pin,

74H21

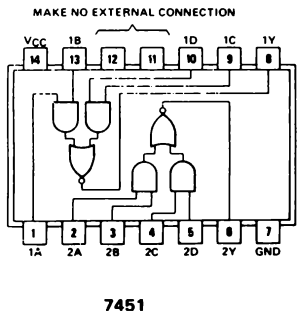
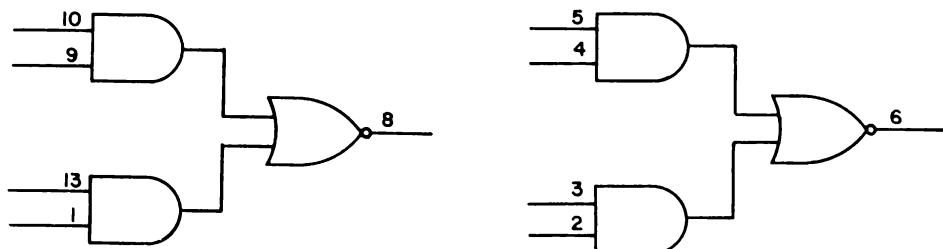
che nella rappresentazione delle porte,

la H di 74H21 ha lo stesso significato della H di 74H11: il chip è ad alta potenza (high power) e può operare ad una velocità di commutazione più elevata del chip 7421 normale.

7427, PORTA NOR A TRE INGRESSI (THREE INPUT NOR GATE)

Il chip 7427, tripla porta NOR positiva a 3 ingressi

7427

contiene tre porte NOR a 3-ingressi *indipendenti*,

7451, PORTA AND-OR INVERTITORE (AND-OR INVERT GATE)

L'ultimo chip di questo capitolo è il 7451, doppia porta AND-OR INVERTITORE a 2-vie, 2-ingressi

7451

che contiene una coppia di porte AND-OR INVERTITORE 2-wide, 2-input.

SOTTOFAMIGLIA TTL

Le iniziali *TTL* significano logica transistor-transistor (transistor-transistor logic) e danno il nome ad una delle molte *famiglie della logica digitale.* La TTL è una delle famiglie oggi più in uso, e questo in conseguenza dei vantaggi che offre: basso costo, capacità di operare ad alta velocità, mancanza di rumore di fondo, possibilità di pilotare fino a dieci altri chip TTL e la vasta gamma di tipi di funzioni disponibili in chip prodotti da diverse industrie. Non vogliamo adesso discutere dettagliatamente delle caratteristiche base delle porte TTL, ma d'altronde crediamo che abbiate già delle conoscenze tali da poter capire il motivo per cui talvolta, quando facciamo la descrizione di un chip, usiamo le iniziali H, L, S, e LS.

Le principali caratteristiche di ogni porta TTL sono: *il tempo di propagazione della porta, la potenza dissipata per porta e la frequenza massima* che è possibile usare con quella porta. Il tempo di propagazione è il tempo necessario affinchè un segnale logico possa compiere il suo lavoro, in un dispositivo o in una serie di dispositivi, per formare una "catena logica"; infatti gli stati logici non passano istantaneamente attraverso una porta, ma esiste un tempo finito caratteristico del passaggio di un impulso di clock attraverso una certa porta. Questo tempo o delay, è detto *delay di memorizzazione, di salita, di caduta o di commutazione,* ed è in funzione della composizione fisica del semiconduttore. Per una normale porta TTL , il tempo di propagazione è di circa 10 ns, cioè 0,00000001 s. La potenza dissipata è il consumo di energia di una porta durante il normale funzionamento. Più energia viene consumata e più attenzione dobbiamo mettere nel dissipare il calore che produce un sistema contenente molti circuiti integrati. Parlando della frequenza massima,si può dire che più basso è il tempo di propagazione, più alta è la frequenza che può essere usata. Comunque parleremo più tardi della frequenza, quando avrete sviluppato una maggior capacità di capire cosa significhi.

Uno degli scopi più importanti della tecnologia dei semiconduttori è quello di fabbricare porte che abbiano dei tempi di propagazione molto bassi, pochissima dissipazione di potenza per porta, e la più alta frequenza massima possibile. Nella famiglia TTL, se volete dei tempi di propagazione molto bassi, dovete usare i chip *Schottky TTL* che vengono contraddistinti con una S dopo il 74; quindi il 74S00, 74S04 e 74S10 sono dei chip che hanno un tempo di propagazione di soli 3 ns. Quindi dieci di queste porte messe in serie hanno un ritardo di propagazione di 30 ns, tempo che è molto basso. Però per ottenere dei ritardi così contenuti dovete pagare una penalità: aumenta di molto infatti, la potenza dissipata per porta, come mostrato nella tabella 4.

Tabella 4 - Riassunto delle caratteristische della sottofamiglia TTL.

Sottofamiglia TTL	Abbrev.	Tempo di propagazione per porta	Potenza dissipata per porta	Frequenza massima
TTL Normale	-	10 ns	10 mW	35 MHz
TTL Alta potenza	H	6 ns	22 mW	50 MHz
TTL Bassa potenza	L	33 ns	1 mW	3 MHz
TTL Schottky	S	3 ns	19 mW	125 MHz
TTL Schottky Bassa potenza	LS	10 ns	2 mW	45 MHz

Se dunque avete bisogno di una bassa dissipazione di potenza per porta, il che significa che le porte hanno bisogno di poca corrente, potete usare sia i TTL bassa-potenza che TTL Schottky a bassa-potenza che hanno rispettivamente una dissipazione di potenza per porta di 1 e 2 mW. Naturalmente bisogna pagare un prezzo per ottenere questa bassa dissipazione: un tempo di propagazione molto lungo, come si può vedere dalla tabella 4. I chip a bassa-potenza vengono prodotti da un grande numero di fabbricanti e sono quindi facilmente reperibili. Questi chip vengono identificati da una L o LS posta dopo il 74, come, ad esempio: 74L00, 74L42, 74L04, 74LS05 e 74LS155.

Il vostro microcomputer MMD-1 funziona a 750 kHz, i! che significa che un clock di 750 kHz guida il chip 8080A e che ogni ciclo di clock ha una durata di 1,333 ns, quindi il Microcomputer MMD-1 non è un microcomputer veloce. I chip TTL usati nella circuiteria del minicomputer e i chip TTL che usate per interfacciare i circuiti, potrebbero in via di principio, lavorare ad una velocità più alta. Il chip più lento di tutti quelli che compongono il microcomputer è la memoria cancellabile/programmabile a sola lettura, la cui sigla è 1702A/8702A, per la quale la frequenza massima è di circa 1 MHz, mentre il resto del microcomputer potrebbe operare a delle velocità di clock più alte di 2 MHz.

Potrete trovare una trattazione più ampia delle caratteristiche dei TTL normali e delle sottofamiglie nel testo *TTL Cookbook* [Horvard W. Sams & Co. Inc, Indianapolis, 1974] di Donald E. Lancaster.

FAN IN E FAN OUT

Fan in e *Fan out* sono delle caratteristiche dei chip digitali molto importanti. La loro definizione è la seguente,

Fan in Specifiche di carico richieste per un ingresso digitale ad un circuito integrato. Per la famiglia logica TTL le specifiche di ingresso sono normalizzate, per i TTL normali, al valore 1. Un Fan in di 1 corrisponde a 1,6 mA.

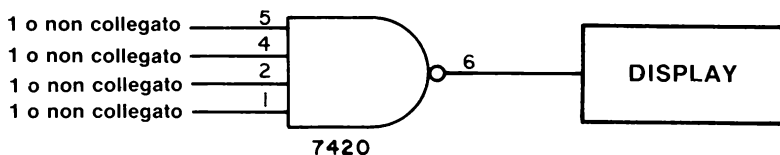
Fan out Capacità di pilotaggio in uscita, di un'uscita digitale di un circuito integrato. Per la famiglia logica TTL, la capacità di pilotaggio di uscita ha il valore 10, cioè 16 mA, allo stato logico 0.

In altre parole, l'uscita di un chip TTL normale può pilotare l'ingresso di altri dieci chip TTL normali. Un chip con un fan-out di 10 può pilotare dieci chip, ciascuno dei quali abbia un fan-in di 1. Siccome la corrente di un fan-in di 1 è 1,6 mA, allora, la capacità di pilotaggio di una uscita TTL è di 16 mA, cioè dieci volte 1,6 mA. Alcuni chip, detti buffer o driver, hanno una capacità di pilotaggio più alta, che si aggira di solito su un fan-out di.30, corrispondente a 48 mA. I chip TTL a bassa potenza hanno un fan-in di circa 0,1 e vengono spesso usati con i chip LSI come i microprocessori.

INGRESSI NON COLLEGATI

In questo capitolo abbiamo visto che ci sono dei chip a porte con due, tre quattro o anche otto ingressi, ma in alcuni circuiti, potreste non aver bisogno di tutti questi ingressi e quindi avreste il problema di cosa fare di quelli non collegati. Per tutta la famiglia TTL si applica la seguente regola molto importante;

Un ingresso non collegato di un circuito integrato viene posto generalmente all'1 logico.

Ci sono alcune eccezioni, come nel chip 8212, che necessita di un ingresso di clear tramite un resistore pull-up. Per esempio, l'uscita del seguente circuito,

deve essere 0 logico, poichè il 7420 è una porta NAND a 4-ingressi e lo stato unico si ha solo quando tutti gli ingressi sono all'1 logico. Occorre mettere a 1 tutti gli ingressi non collegati altrimenti, se il display fosse un indicatore a LED, non si accenderebbe mai. Quindi sono due le regole che dovrete sempre ricordare:

Se avete dei chip della serie 7400, è possibile lasciare degli ingressi non collegati, ognuno di questi deve però essere posto all'1 logico.

Se state cablando un circuito digitale su un circuito stampato o su una piastra, dovete collegare tutti gli ingressi non utilizzati ai +5V (per mezzo di un resistore pull-up da 1 kΩ) o alla massa. Un solo resistore pull-up può essere usato per più ingressi all'1 logico.

Il collegare allo 0 o 1 logico gli ingressi non utilizzati dei chip TTL dei circuiti digitali è un'ottima pratica costruttiva.

NUMERI DI CHIP E CODICI DI DATA

Di solito ogni circuito integrato ha una identificazione data direttamente dal costruttore. Sul dorso di ogni chip c'è il suo numero, il *codice di data* e altre informazioni occasionali. Il codice di data riporta l'anno e la settimana dell'anno nella quale il chip è stato costruito e provato.

Un codice di data 7106 indica che il chip è stato costruito e provato nella sesta settima del 1971. State attenti ai chip fabbricati nel 1974, infatti il numero 7410 potrebbe voler dire circuito 7410, cioè tripla porta NAND a 3-ingressi, oppure che il chip in questione è stato costruito e provato nella decima settimana del 1974. Dovreste riuscire a vedere alcuni codici di data nella Figura 10-1. Il codice di data del chip 7400 a sinistra, è 7319; quello del chip 7476 è 7232; quello del chip 74150 di 24 pin, è 7312; e quello del chip AY-5-1012 di 40 pin è 7308.

Oltre al codice di data, troverete su ogni chip,

Il nome del costruttore, il suo simbolo o le sue iniziali
Il numero di serie 7400 o qualche altro numero relativo al chip.

Per voi è importante solo il numero di serie 7400 (o serie 5400, col 5 che sostituisce il 7). I maggiori costruttori di chip della serie 7400 sono:

ITT Semiconductors (ITT)
National Semiconductor (NS)
Sprague Electric Company (Sprague)
Stewart Warner Corp. (SW)
Texas Instruments (ti)
Signetics (S)
Siemens - AEG (S)
Motorola (M)
SGS - Ates (SGS - ATES)
Harris Semiconductor (H)
Fairchild Semiconductor (F)
Raytheon Semiconductor (Ray)

I chip microprocessori vengono venduti da un certo numero di costruttori, ma la famiglia dei chip più usata, la famiglia 8080, è delle seguenti aziende: Intel Corporation, Siemens - AEG, National Semiconductor, NEC, Texas Instruments, AMD e Hitachi. Il microprocessore 8080, ha una versione maggiorata, l'8080A, che è acquistabile ad un prezzo un po' più alto.

Per comprendere meglio l'identificazione di un chip della serie 7400, osservate i seguenti esempi:

- *Sprague 7106 US7400A*

  Questo è un 7400, quadrupla porta NAND a 2-ingressi, costruito nella 6ª settimana del 1971 dalla Sprague Electric Company.

- *SW 7400-N 20231 7216*

  Anche questo è un 7400, quadrupla porta NAND, a 2-ingressi, ma è costruito dalla Stewart Warner Corporation. Il codice di data si riferisce alla 16ª settimana del 1972. Il numero 20231 è poco importante.

- *NS 351 DM7489 N*

  Questo è un 7489, memoria a lettura/scrittura a 64 bit costruito dalla National Semiconductor nella 51ª settimana del 1973.

- *7324 SN 7440 N e "mappa del Texas"*

  Questo è un 7440, doppio buffer NAND positivo a 4-ingressi costruito dalla Texas Instruments nella 24ª settimana del 1973.

DIMENSIONI DEL CONTENITORE DUAL-IN-LINE

Le dimensioni, sia in pollici che in millimetri, del contenitore dual-in-line, sono illustrate in questa pagina e nella seguente. I dati sono riportati per gentile concessione della Texas Instruments Incorporated.

TTL INTEGRATED CIRCUITS MECHANICAL DATA

J ceramic dual-in-line packages (inch dimensions, see page 55 for metric dimensions)

These hermetically sealed, dual-in-line packages consist of a ceramic base, ceramic cap, and a 14-, 16-, or 24-lead frame. The circuit bar is alloy-mounted to the base and hermetic sealing is accomplished with glass. The packages are intended for insertion in mounting-hole rows on 0.300-inch or (0.600-inch) centers. Once the leads are compressed and inserted, sufficient tension is provided to secure the package in the board during soldering. Tin-plated ("bright-dipped") leads (−00) require no additional cleaning or processing when used in soldered assembly.

14-PIN J CERAMIC DUAL-IN-LINE PACKAGE OUTLINE

Falls Within JEDEC TO-116 and MO-001AA Dimensions

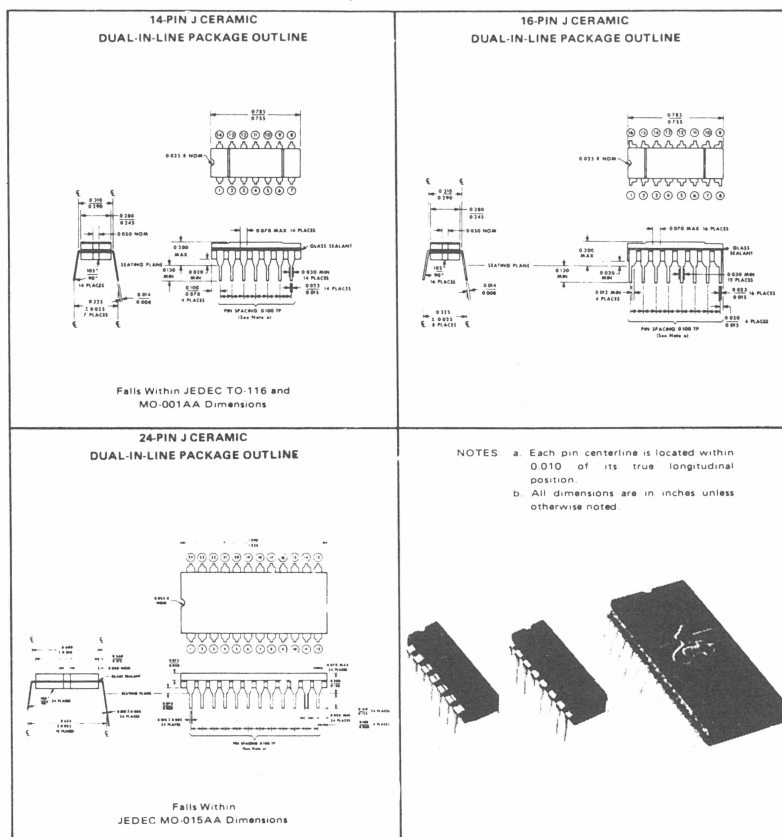
16-PIN J CERAMIC DUAL-IN-LINE PACKAGE OUTLINE

24-PIN J CERAMIC DUAL-IN-LINE PACKAGE OUTLINE

Falls Within JEDEC MO-015AA Dimensions

NOTES: a. Each pin centerline is located within 0.010 of its true longitudinal position.
b. All dimensions are in inches unless otherwise noted.

TEXAS INSTRUMENTS INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

51

TTL INTEGRATED CIRCUITS MECHANICAL DATA

N plastic dual-in-line packages (metric dimensions, see page 52 for inch dimensions)

These dual-in-line packages consist of a circuit mounted on a 14-, 16- or 24-lead frame and encapsulated within an electrically nonconductive plastic compound. The compound will withstand soldering temperature with no deformation and circuit performance characteristics remain stable when operated in high-humidity conditions. These packages are intended for insertion in mounting-hole rows on 7.62-mm (or 15.24-mm) centers. Once the leads are compressed and inserted, sufficient tension is provided to secure the package in the board during soldering. Silver-plated leads (—00) require no additional cleaning or processing when used in soldered assembly.

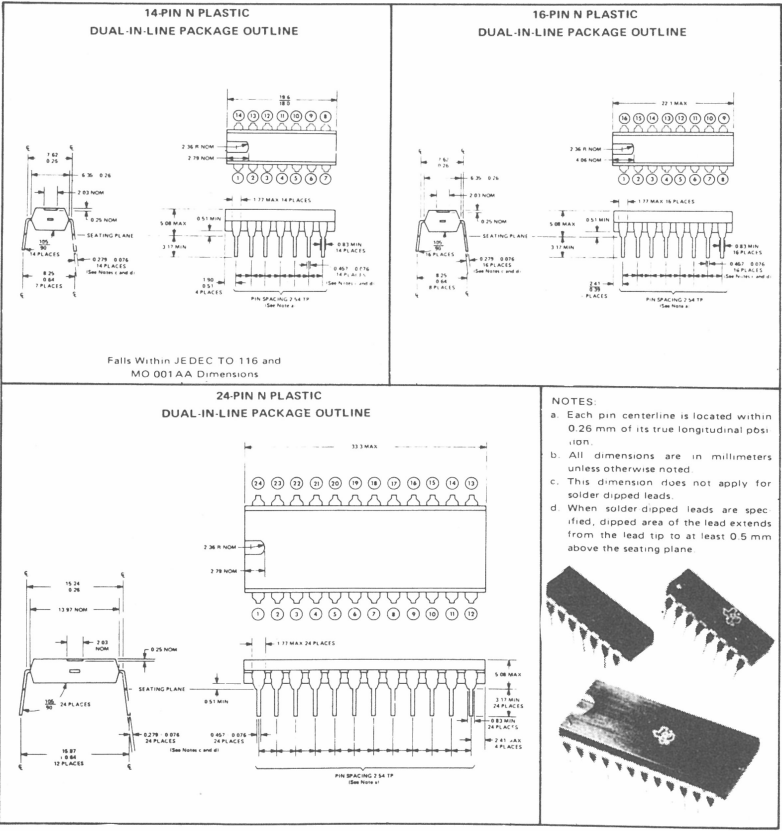
NOTES:

a. Each pin centerline is located within 0.26 mm of its true longitudinal position.

b. All dimensions are in millimeters unless otherwise noted.

c. This dimension does not apply for solder dipped leads.

d. When solder dipped leads are specified, dipped area of the lead extends from the lead tip to at least 0.5 mm above the seating plane.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

ALTRI CONSIGLI UTILI

Nel capitolo 9 vi abbiamo dato dei consigli e dei suggerimenti sull'uso dei fili, del breadboard e delle funzioni ausiliarie; adesso vorremmo fornirvi dei suggerimenti sull'uso degli utensili, dei circuiti integrati e di altri componenti elettronici.

Utensili

- Per fare tutti gli esperimenti di questi capitoli sono necessari solo tre utensili,

  Un paio di pinze da laboratorio [di solito usiamo le XCELITE da 5 pollici con lame piatte, pinza a punta zigrinata con molla, n^0 72 CG]

  Un cacciavite [va bene un qualsiasi cacciavite economico della lunghezza di 4 o 5 pollici e con la punta di 1/8 di pollice]

  Uno spelafili [preferiamo l'XCELITE, spelafili n^0 101-S]

- Le pinze vengono usate per raddrizzare le estremità del filo adoperato per collegare sulla scheda i circuiti integrati.

- Le pinze vengono usate anche per raddrizzare o piegare nella posizione voluta i pin del circuito integrato.

- Sempre le pinze, vengono usate per piegare in modo voluto le estremità dei resistori, dei condensatori, per poterli facilmente inserire nella piastra senza saldature.

- Il cacciavite è un ottimo utensile per togliere i circuiti integrati. Basta inserire il cacciavite nella tacca centrale dell'SK-10 o SK-50 e fare lentamente forza dai terminali di centro. Per evitare che i pin si stortino bisogna far forza da entrambe le estremità del chip.

- Il cacciavite viene usato anche per togliere le funzioni ausiliarie Outboard dai bordi della piastra. Per evitare di danneggiare la parte inferiore degli Outboard vi consigliamo di rivestire la punta del cacciavite con del nastro isolante.

- Le pinze servono anche per togliere i fili dai terminali del breadboard. Abbiamo notato che le pinze servono anche quando c'è bisogno di aggiungere qualche filo alla "giungla" di cavi già esistenti.

Componenti elettronici

- Per fare gli esperimenti di questa serie di capitoli avrete bisogno anche di un certo numero di resistori:

Resistori pull-up da 1000Ω per le uscite a "open collector" dei circuiti integrati (ne necessitano circa cinque).

Resistori limitatori di corrente da 220Ω per i diodi emettitori di luce e per i display a LED a sette segmenti (ne necessitano circa sette).

Resistori di valore diverso, tra 1000Ω e 1 MΩ, da usare per determinare le costanti di tempo RC.

Per queste applicazioni vanno benissimo i resistori al carbone da 1/4 W oppure da 1/8 W.

- Condensatori di diversi valori, tra 20 pF e 5 nF, che servono per determinare le costanti di tempo RC nei timer e nei multivibratori monostabili.

- Diodi (LED) emettitori di luce rossa, verde e gialla che costano abbastanza poco e sono prodotti da molti fabbricanti.

- A noi piace il display a sette segmenti a LED Opcoa SLA-1, che ha parecchie sorgenti, ed è il display che viene usato con l'Outboard LR-4. Il display 5082-7730 della Hewlett Packard è migliore, ma costa di più dell'Opcoa SLA-1 e viene anch'esso usato con l'Outboard LR-4.

- La serie 5082-7300 di display e indicatori numerici ed esadecimali della Hewlett-Packard è molto bella. Ha al suo interno un latch e un decoder/driver e quindi è superiore allo stesso Outboard LR-4 che contiene solo un decoder/driver e non il latch.

Circuiti integrati

- I circuiti integrati da 14 e 16 pin possono di solito venir inseriti direttamente sulla piastra SK-10, però prima assicuratevi che i pin non siano piegati, ma ben verticali. Se eventualmente doveste riportare in posizione verticale qualche pin, usate le pinze od una superficie piatta.

- I pin di alcuni circuiti integrati hanno le estremità appuntite, mentre altri sono completamente piatti. Evidentemente quelli appuntiti sono più facilmente inseribili nel breadboard.

- I circuiti integrati più cari o i chip che hanno più di 24 pin, dovrebbero essere messi su degli *zoccoli per circuiti integrati*, prima di venire inseriti sulla piastra. Gli zoccoli vengono forniti da molti costruttori.

- Quando fosse necessario inserire e togliere più volte un circuito integrato da un certo punto della piastra, è meglio usare degli *zoccoli con forza di inserimento zero.*

  Questo è il caso, per esempio, di quando bisogna fare dei test su una serie di chip.

- I circuiti integrati della serie 7400 della famiglia TTL sono oggi relativamente poco rari; acquistateli con giudizio e risparmierete denaro.

- Comprate solo circuiti integrati dual-in-line package (DIP). I circuiti integrati di formato piatto (flat-pack), che tra l'altro non si trovano facilmente, sono usabili con difficoltà sulla piastra SK-10.

- Tutti i circuiti integrati hanno stampato sulla parte superiore il tipo di chip e il codice di data; alcuni hanno, sul lato inferiore, altri numeri. Non cancellate, mutilate, pasticciate questi numeri, ma anzi inventatevi un sistema per poter facilmente identificare i chip il cui numero sia diventato illegibile. Noi abbiamo scoperto che il BUGBACK® commercializzato dalla E. & L. Instruments, Inc., (per l'Italia Microlem S.p.A.) è molto efficiente, ma dei piccoli barattoli vanno molto bene per tener divisi i diversi tipi di chip.

- *Quando un circuito integrato brucia, di solito la causa è un errore nei collegamenti di alimenatzione alla piastra, infatti se collegate i +5V all'ingresso di MASSA e la MASSA all'ingresso Vcc, bruciate tutti i chip. Quindi, per favore, state molto attenti!*

- A volte il modo di riporre e conservare i chip può creare dei problemi, infatti non vanno tenuti ammucchiati uno sopra l'altro perchè i pin potrebbero danneggiarsi. Noi vi consigliamo di usare del polistirolo espanso con delle etichette o delle scatolette di plastica etc.

- Maneggiate con estrema cura i chip metal-oxid semiconductor (MOS) come l'8080A e quasi tutti i chip di memoria. Se dovete trasportarli inseriteli su una base di materiale espanso conduttore o avvolgeteli in fogli di alluminio e durante l'inverno, quando in casa o in ufficio possono formarsi delle cariche elettrostatiche, state attenti quando li collegate all'8080A, perchè queste cariche potrebbero distruggerli.

- Quando un chip costa più di 40$ diventa oggi una grossa spesa, ma ricordate che nel 1975 un 8080 acquistato singolarmente costava 360$.

- Identificare la funzione di tutti i pin di un circuito integrato è un aspetto importante dell'elettronica digitale. Scoprirete che i seguenti manuali di specifiche dati dai costruttori sono molto utili:

  "The TTL Data Book for Design Engineers"
  Texas Instruments Italia
  Nucleo Industriale
  02015 CITTADUCALE

  "Signetics Digital, Linear, and MOS Handbook"
  Philips Elcoma
  Piazza IV Novembre - 20124 MILANO

  "Digital Integrated Circuits"
  National Semiconductor S.r.l.
  Via A. Mario, 26 - 20159 MILANO

  "Intel 8080 Micrcomputer Systems User's Manual"
  Eledra 3S S.p.A.
  Viale Elvezia, 18 - 20154 MILANO

Noi pensiamo che questi manuali debbano essere le vostre "Bibbie" se volete costruire dei nuovi circuiti digitali.

- I prezzi dei circuiti integrati variano molto rapidamente, ma quelli che riportiamo qui sotto possono darvi un'idea del loro costo all'inizio del 1976 in America.

Bassa scala di integrazione (SSI)	prezzo cadauno *
porte semplici	da 0,16$ a 0,30$
flip-flop	da 0,45$ a 0,80$
Media scala di integrazione (MSI)	
Contatori	da 0,70$ a 1,40$
Decodificatori	da 1,08$ a 1,50$
Multiplexer	da 1,10$ a 1,40$
Registri di shift	da 1,20$ a 2,25$
Decodificatori/driver	da 1,10$ a 1,15$
Unità aritmetico/logiche	da 1,15$ a 3,55$
Piccole memorie a lettura/scrittura	da 3,00$ in su
Larga scala di integrazione	
Microprocessor 8080	39,95$
AY-5-1013 UART	6,95$
Memoria 2102	2,95$
Memoria 8111	7,95$
Generatore di impulsi di clock 8224	12,95$
EPROM 1702A	da 7,00$ a 15,00$

* Prezzi per hobby, surplus o saldi.

INTRODUZIONE AGLI ESPERIMENTI

I seguenti esperimenti vi permetteranno di provare le proprietà di un certo numero di chip della serie 7400. Questi chip sono economici e abbastanza durevoli. Se il vostro chip si scalda, probabilmente avete collegato male l'alimentazione.

Gli esperimenti che farete possono essere così riassunti:

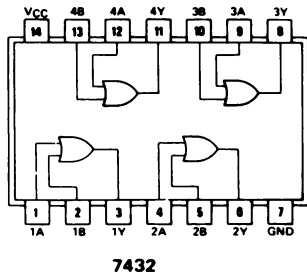
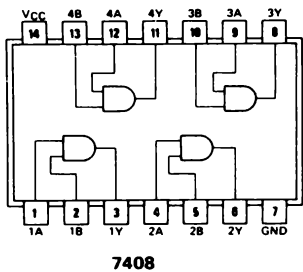
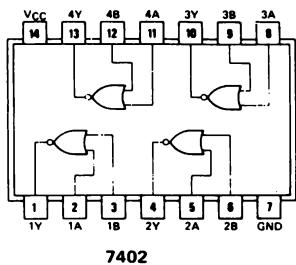
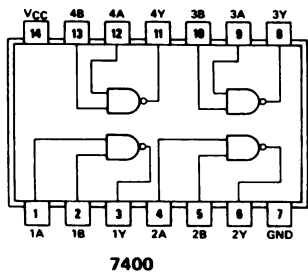
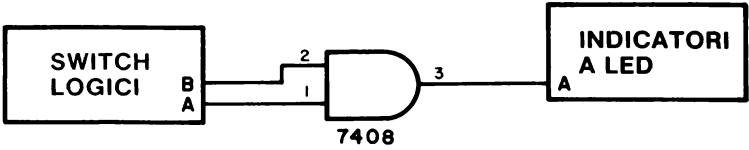
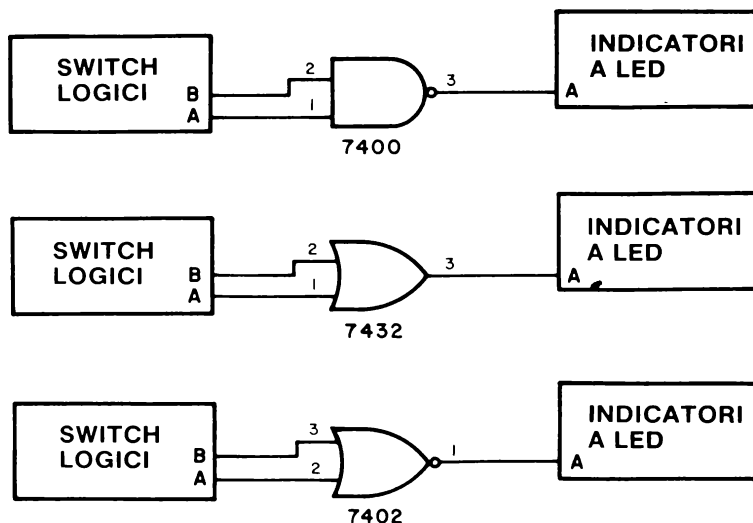
Esperimento N.	Commento
1	Trovare la tabella della verità per alcuni dei chip più comuni della serie 7400: 7400, 7402, 7408 e 7432.
2	Trovare la tabella della verità per una porta NAND a 3-ingressi.
3	Determinare lo stato logico dell'uscita di una porta NAND a 4-ingressi quando non tutti i pin di ingresso della porta sono collegati ad uno stato logico specifico.
4	Trovare la tabella della verità di tre porte OR della serie 7400: 7402, 7432 e 7486.

Esperimento N.	Commento
5	Mostrare come si usa il microcomputer per trovare la tabella della verità per diversi tipi di porte logiche a 2-ingressi.
6	Trovare la tabella della verità per quattro porte della serie 7400: 7411, 74H21, 7427 e 7430.
7	Mostrare l'uso della porta OR-Esclusivo 7486 come invertitore controllabile.
8	Trovare la tabella della verità della porta AND-OR-INVERTITORE 7451.

ESPERIMENTO N. 1

Scopo

Lo scopo di questo esperimento è di determinare sperimentalmente la tabella della verità di quattro porte a 2-ingressi della serie 7400: la 7400, porta NAND a 2-ingressi, la 7402, porta NOR a 2-ingressi, la 7408, porta AND a 2-ingressi, e la 7432, porta OR a 2-ingressi.

Configurazioni dei pin dei circuiti integrati

Schemi dei circuiti

Passo 1

Dovete trovare la tabella della verità delle quattro porte illustrate negli schemi, seguendo l'ordine con cui sono riportate. *Osservate che non sono stati illustrati i collegamenti di alimentazione e che viene provata sperimentalmente solo una delle quattro porte del chip.*

Passo 2

Su un broadboard senza saldature SK-10 separato, inserite o un Outboard LR-25 oppure un Outboard con switch logici LR-2 e un Outboard con indicatori a LED LR-6. Mettete l'Outboard LR-2 sul lato opposto a quello dove si trova l'Outboard LR-6 inserite l'Outboard di *alimentazione* LR-1, il tutto come mostrato nella Figura 9-9. Se invece usate l'Outboard LR-25, non avete bisogno dell'Outboard LR-1.

Passo 3

Riguardate i consigli e i suggerimenti che vi abbiamo dato in questo capitolo, specialmente quelli che riguardano i circuiti integrati e poi leggete subito i seguenti:

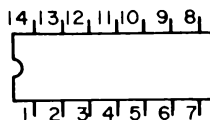
- Staccate l'alimentazione dalla piastra prima di collegare i circuiti integrati o prima di fare dei cambiamenti nel circuito.
- Controllate se sapete dove si trova il pin 1 di ogni circuito perchè se li collegate al contrario potrebbero bruciare.
- Ricordate che per funzionare ogni chip ha bisogno di due collegamenti di alimentazione, +5V MASSA e che questi collegamenti non appaiono sugli schemi.
- Non mettere i pin dei circuiti sul lato opposto del breadboard in corrispondenza dei pin degli Outboard.

- Controllate bene e identificate il numero del chip e il codice di data riportato sul dorso del chip stesso.

- *Prima di ogni altra cosa,* eseguite i collegamenti tra i circuiti integrati e i terminali di alimentazione del breadboard; dopo di che accertatevi che sia stato eseguito correttamente. È più facile non commettere errori nei collegamenti di alimentazione se non ci sono già altri fili sulla piastra.

- Dopo aver collegato ogni circuito digitale, accertatevi che non ci siano errori.

Passo 4

Il circuito integrato 7402, porta AND a 2-ingressi, è un chip a 14 pin che può avere o una tacca semicircolare ad una estremità o un piccolo incavo sul contenitore plastico nella parte opposta rispetto al pin 1,

inserite il chip 7408 nel breadboard come mostrato nella Figura 10-2, *ma non mettete in opposizione, uno contro l'altro, i pin del circuito con quelli di un qualsiasi Outboard che state usando.* Se lo fate, il circuito potrebbe non funzionare correttamente o addirittura bruciare. Osservate la Figura 10-2, la tacca semicircolare si trova nella parte in alto del chip.

Passo 5

Trovate il pin 1 del circuito integrato e quindi, in base ad esso, il pin 7 e collegatelo a MASSA (GND) come mostrato nella Figura 10-2. Trovate il pin 14 del chip e collegatelo ai +5V come mostrato nella Figura 10-2.

Passo 6

Fate gli altri collegamenti tra il chip e gli Outboard degli switch logici e degli indicatori a LED. Usate lo switch che vi risulta più comodo e uno qualsiasi dei quattro indicatori dell'Outboard LR-6 (o uno degli otto dell'Outboard LR-25).

Passo 7

Collegate l'alimentazione e mettete gli switch logici al valore 1. L'indicatore a LED collegato al pin 3 del chip 7408, dovrebbe accendersi indicando che l'uscita della porta AND a 2-ingressi è all'1 logico.

Portate gli switch logici A e B allo 0 logico. Adesso la luce dovrebbe spegnersi.

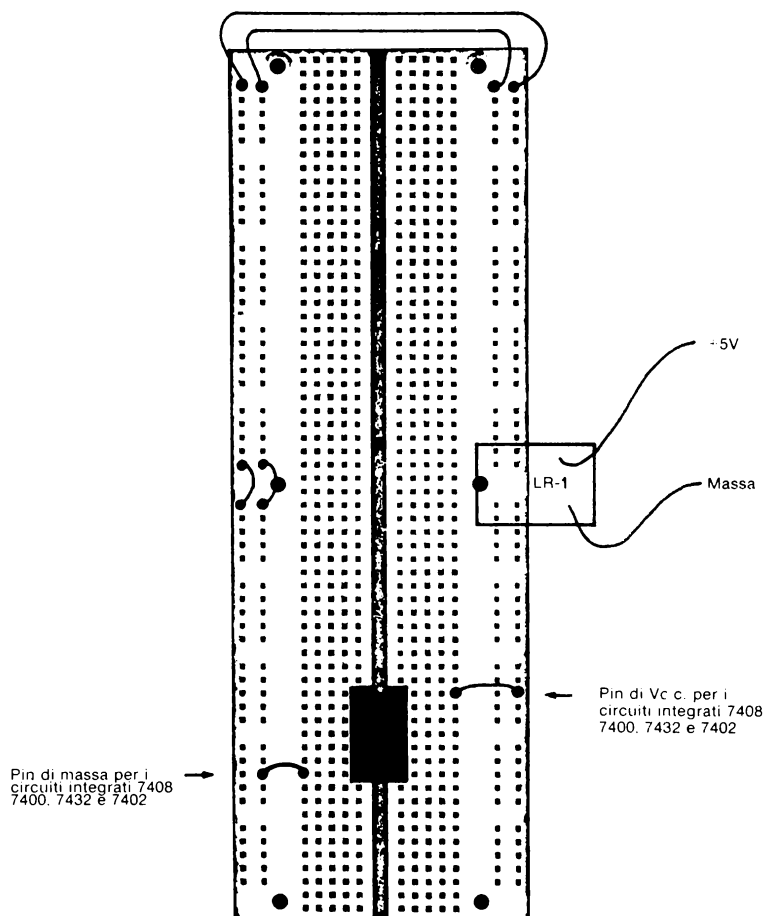



Figura 10-2. Schema dei collegamenti che mostra la dislocazione dei pin di ingresso V_{CC} e MASSA dei circuiti integrati 7400, 7402, 7408 e 7432. Osservate che la tacca semicircolare è posta verso l'alto.

Passo 8

Commutando in tutti i modi possibili gli switch, compilate la seguente tabella della verità per una porta AND a 2-ingressi

B	A	Q
0	0	
0	1	
1	0	
1	1	

La tabella della verità completa e corretta è data nel capitolo 7 ed è riportata qui sotto:

B	A	Q
0	0	0
0	1	0
1	0	0
1	1	1

Solo quando A e B sono all'1 logico, l'uscita è all'1 logico, che corrisponde all'indicatore a LED acceso.

Passo 9

Staccate l'alimentazione dal breadboard e togliete con cautela il chip 7408 servendovi di un cacciavite. Sostituitelo con un chip 7400, porta NAND a 2-ingressi e non cambiate i collegamenti. Ricollegate l'alimentazione e verificate che la porta NAND a 2-ingressi ha la seguente tabella della verità:

B	A	Q
0	0	1
0	1	1
1	0	1
1	1	0

Questa tabella della verità era già stata discussa nel capitolo 7. Osservate che lo stato unico si ha quando sia A che B sono all'1 logico; e in questo caso, lo stato di uscita unico è allo 0 logico.

Passo 10

Staccate l'alimentazione dal breadboard e sostituite il chip 7400 con un chip 7432, porta OR a 2-ingressi. Ricollegate l'alimentazione e trovate la tabella della verità per la porta OR 7432.

B	A	Q
0	0	
0	1	
1	0	
1	1	

L'esatta tabella della verità di una porta OR positiva a 2-ingressi è stata data anche nel capitolo 7:

B	A	Q
0	0	0
0	1	1
1	0	1
1	1	1

Passo 11

Scollegate l'alimentazione del breadboard e rimpiazzate il chip 7432 col chip 7402, porta NOR a 2-ingressi. Cambiate i collegamenti come mostrato nello schema; adesso l'uscita della porta NOR è il pin 1 e non più il 3. Ricollegate l'alimentazione e verificate se la seguente tabella della verità è quella giusta:

B	A	Q
0	0	1
0	1	0
1	0	0
1	1	0

Passo 12

Riguardate la configurazione dei pin dei quattro circuiti integrati; potreste provare il comportamento di ognuna delle altre tre porte di ogni chip perché ognuna di esse è indipendente e se una porta si brucia, si può usare una delle tre che rimangono.

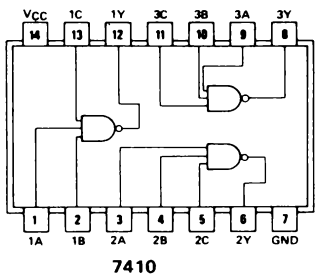
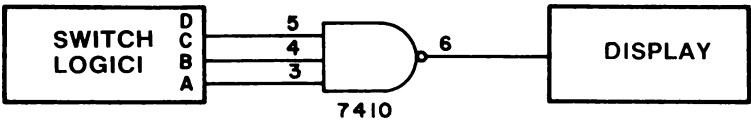
Passo 13

Togliete l'alimentazione +5V da uno dei chip che avete provato, come, per esempio, il chip 7402. Potete adesso riprodurre la tabella della verità di questo chip? La risposta è: assolutamente *no*; infatti il chip funziona solo se ha l'alimentazione collegata ai pin 7 e 14 e se la tensione è superiore a quella prescritta, può bruciare tutto.

ESPERIMENTO N. 2

Scopo

Lo scopo di questo esperimento è determinare la tabella della verità di una porta NAND positiva a 3-ingressi di un circuito integrato 7410.

Configurazione dei pin del circuito integrato

Schema del circuito

Passo 1

Il "display" mostrato nello schema può essere sia un singolo indicatore a LED che un display a sette segmenti o un latch/display collegato come un display zero/uno:

Se usate un display zero/uno, l'uscita della porta apparirà come "0" o "1". Nell'Esperimento N. 1, il display era costituito dall'indicatore a LED A.

Passo 2

Inserite il circuito integrato 7410 nel breadboard e identificate in pin 1. Fate i collegamenti di alimentazione tra il chip, +5V e MASSA. Fate il collegamento tra il chip, gli switch logici e il display e infine controllate che sia tutto a posto.

Passo 3

Alimentate il breadboard e commutate gli switch logici per determinare sperimentalmente i valori di Q nella sottostante tabella della verità:

C	B	A	Q
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Confrontate quanto avete osservato con la tabella della verità esatta che abbiamo già discusso nel capitolo 7:

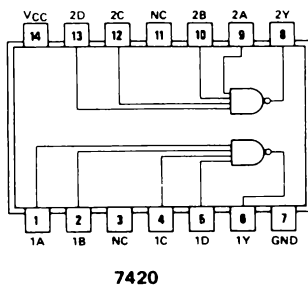
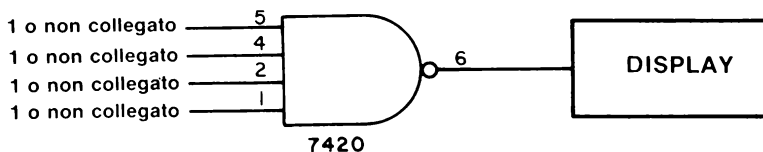
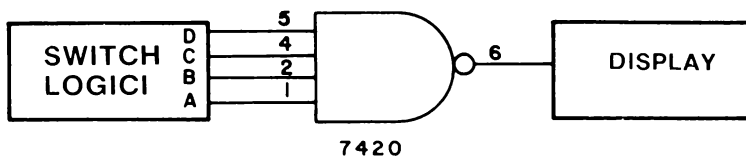
C	B	A	Q
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

C'è un solo stato al quale corrisponde l'uscita 0; questo è lo stato unico della porta e si ha quando A, B e C sono all'1 logico.

ESPERIMENTO N. 3

Scopo

Lo scopo di questo esperimento è di determinare lo stato logico dell'uscita di una porta NAND positiva a 4-ingressi, 7420, quando i pin d'ingresso della porta non sono collegati ad uno specifico stato logico.

Configurazione dei pin del circuito integrato

Schemi dei circuiti

Passo 1

Staccate l'alimentazione dal breadboard e collegate questo circuito nel quale la porta 7420 è legata a quattro switch logici ed a un display. Rialimentate il breadboard e verificate se la seguente tabella della verità per una porta NAND a 4-ingressi, è corretta:

D	C	B	A	Q
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Passo 2

Staccate i quattro collegamenti ai quattro switch logici. Adesso il vostro circuito sul breadboard è identico al circuito in basso della Figura precedente. Ridate alimentazione. Che uscita logica vedete?

Noi vediamo che lo stato dell'uscita del chip 7420, quando nessuno degli ingressi è collegato, è allo 0 logico.

Cosa deducete circa il comportamento dei pin di ingresso non collegati di un chip 7420? E questa conclusione potrebbe essere vera per tutti gli ingressi non collegati dei circuiti integrati della serie 7400?

L'importante conclusione che dovreste aver tratto è che un ingresso non collegato del chip 7420, è allo stato logico 1!

Discussione

In genere, per la serie 7400, il risultato che si può osservare quando gli ingressi non sono collegati è che essi sono all'1 logico. Si può quindi enunciare la eseguente regola generale:

Tutti i pin di ingresso non collegati dei chip della serie 7400 sono allo stato logico 1.

Questa regola è vera per tutti i chip TTL della serie 7400 che vedrete in questo Bugbook. In generale è vera per quasi tutti i chip TTL, ma comunque esistono delle eccezioni.

Se state collegando temporaneamente dei circuiti TTL e se volete che qualche pin di ingresso sia sempre all'1 logico, basta che non lo colleghiate - su un circuito stampato o con dei fili su un breadboard - sarebbe meglio collegare tutti gli ingressi non utilizzati ai +5V passando per un resistore pull-up da 1 kΩ. Questa è la procedura che si usa quando si fa della buona progettazione.

Ricordate:

Tutti i pin di ingresso non collegati dei chip della serie 7400 sono allo stato logico 1

I circuiti integrati della serie 7400 hanno delle limitazioni sul numero degli ingressi e delle uscite che possono essere collegati insieme. Di solito non dovete collegare tra loro le uscite (tranne che per alcune porte speciali che abbiano l'uscita three-state o open-collector), ma è possibile collegare una singola uscita agli ingressi di diverse porte. La domanda è: quante porte possono essere collegate ad una singola uscita? C'è un limite e, negli standard della serie 7400, è in relazione ai valori di fan-in e fan-out. Molte porte hanno un fan-out di 10 e un fan-in di 1 e quindi una sola uscita, con un fan-out di 10, può essere collegata a 10 ingressi ognuno dei quali abbia un fan-in di 1. Se un progetto logico è fatto bene, il fan-out non deve mai superare né essere uguale al fan-in, in ogni punto del circuito digitale.

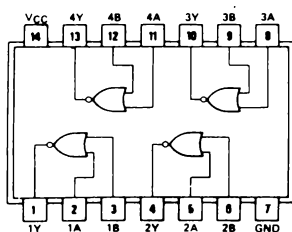
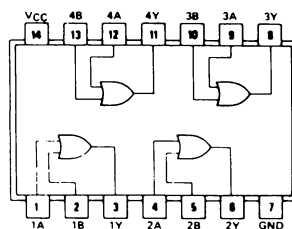
La logica transistor-transistor (TTL) è conosciuta anche come circuit sinking logic. Questo significa che un fan-out di 10 corrisponde ad un assorbimento di corrente di una certa grandezza, in questo caso 16 mA per un'uscita allo stato logico 0. Il fan-in corrisponde alla quantità di corrente che un singolo ingresso ha bisogno quando si trova allo stato logico 0 e un fan-in di 1 corrisponde a 1,6 mA. Quindi la corrente passa dal transistor in ingresso al transistor in uscita della porta o dell'elemento logico precedente. Un'uscita agisce come assorbitore di corrente, mentre un ingresso si comporta come "sorgente" di corrente.

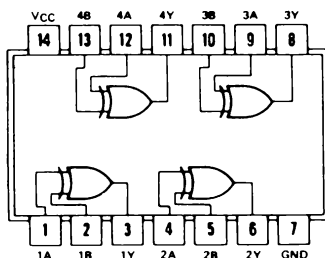
ESPERIMENTO N. 4

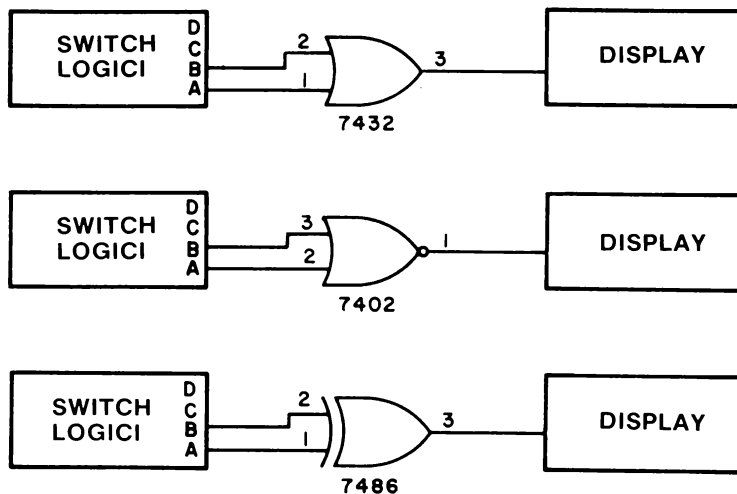
Scopo

Lo scopo di questo esperimento è di determinare la tabella della verità di tre porte di tipo OR della serie 7400:

- porta OR 7432
- porta NOR, o OR NEGATIVO, 7402
- porta OR-Esclusivo 7486

Configurazioni dei pin dei circuiti integrati

7402

7432

7486

Schemi dei circuiti

Passo 1

Avete già determinato la tabella della verità sia per i circuiti integrati 7402 che 7432; in questo esperimento molto semplice, vi chiediamo di confrontare le tabelle della verità di tutti e tre i chip di questo tipo.

Passo 2

Collegate sul breadboard i tre chip e verificate le seguenti tabelle della verità delle tre porte:

Porta OR 7432

B	A	Q
0	0	0
0	1	1
1	0	1
1	1	1

Porta NOR 7402

B	A	Q
0	0	1
0	1	0
1	0	0
1	1	0

Porta OR - Esclusivo 7486

B	A	Q
0	0	0
0	1	1
1	0	1
1	1	0

Discussione

La porta OR-Esclusivo è molto usata nei circuiti digitali perchè esegue addizioni e sottrazioni aritmetiche sui numeri binari; inoltre può essere usata come invertitore controllato anche se, in realtà, non è una vera porta perchè non ha uno stato unico.

ESPERIMENTO N. 5

Scopo

Lo scopo di questo esperimento è di dimostrare l'uso del microcomputer per creare delle tabelle per diversi tipi di porte logiche a 2-ingressi.

Programma

Indirizzo di memoria LO	Byte di istruzione	Codice mnemonico	Descrizione
260	006	MVI B	Muovi in modo immediato il byte seguente nel registro B
261	014	014	Byte di dati che simula i quattro possibili ingressi all'altro pin di una porta a 2-ingressi. La codifica binaria di questo byte di dati è 00001010_2, ma a noi interessano solo i quattro bit meno significativi.
262	076	MVI A	Muovi in modo immediato il byte seguente nell'accumulatore.
263	012	012	Byte di dati che simula i quattro possibili ingressi all'altro pin di una parola a 2-ingressi. La codifica binaria di questo byte di dati è 00001010_2, ma a noi interessano solo i quattro bit meno significativi.
264	240	ANA B	Metti in AND il contenuto del registro B col contenuto dell'accumulatore. [Attenzione: in questa locazione di memoria potrete provare anche le istruzioni XRA B e ORA B].
265	000	NOP	Nessuna operazione. [Attenzione: in questa locazione di memoria potrete provare anche l'istruzione CMA, che complementa il contenuto dell'accumulatore.]
266	323	OUT	Metti in uscita sulla porta 2 il contenuto dell'accumulatore.
267	002	002	Codice di dispositivo della porta 2.
270	166	HLT	Alt del microcomputer.

Passo 1

Caricate sul microcomputer ed eseguite questo programma. Se è un po' di tempo che non usate il microcomputer, per favore fate riferimento al capitolo 8 dove sono già fatti dei programmi che comprendevano istruzioni logiche.

Osservate gli indicatori a LED in uscita sulla porta 2. Dovreste vederci l'uscita binaria 00001000_2. Questo è il risultato logico della operazione di AND tra il contenuto del registro B e il contenuto dell'accumulatore. La relativa tabella della verità è la seguente:

	Registro B	Registro A	Uscita porta 2
Bit 0:	0	0	0
Bit 1:	0	1	0
Bit 2:	1	0	0
Bit 3:	1	1	1

I bit da 0 a 3 sono i bit meno significativi del registro B, dell'accumulatore e della uscita sulla porta 2. Questa è anche la tabella della verità di una operazione logica AND a 2-ingressi.

Passo 2

Cambiate il byte di istruzione all'indirizzo 264 con una istruzione di OR-Esclusivo, 250. Eseguite ancora una volta il programma e osservate l'uscita alla porta 2. Dovreste vedere un'uscita equivalente alla seguente tabella della verità:

	Registro B	Registro A	Uscita porta 2
Bit 0:	0	0	0
Bit 1:	0	1	1
Bit 2:	1	0	1
Bit 3:	1	1	0

Questa è la tabella della verità di una operazione logica di OR-Esclusivo.

Passo 3

Cambiate i byte di istruzione all'indirizzo 264 con una istruzione di OR, 260. Eseguite ancora una volta il programma e notate che l'uscita della porta vi da la tabella della verità per una operazione logica di OR a 2-ingressi ogni volta che eseguite il programma:

	Registro B	Registro A	Uscita porta 2
Bit 0:	0	0	0
Bit 1:	0	1	1
Bit 2:	1	0	1
Bit 3:	1	1	1

Passo 4

Adesso cambiate il byte di istruzione all'indirizzo di memoria 265 con l'istruzione che complementa il contenuto dell'accumulatore, 057. Così facendo, vi siete messi in grado di eseguire una operazione di OR negativo o NOR. Eseguite il programma e dimostrate che l'uscita alla porta 2 è equivalente alla tabella della verità per una operazione logica di NOR a 2-ingressi.

	Registro B	Registro A	Uscita porta 2
Bit 0:	0	0	1
Bit 1:	0	1	0
Bit 2:	1	0	0
Bit 3:	1	1	0

Passo 5

Infine cambiate il byte di istruzione all'indirizzo di memoria 264 con una istruzione di AND, 240, ed eseguite il programma per l'ultima volta. La porta di uscita dovrebbe darvi la tabella della verità dell'operazione logica di NAND a 2-ingressi.

	Registro B	Registro A	Uscita porta 2
Bit 0:	0	0	1
Bit 1:	0	1	1
Bit 2:	1	0	1
Bit 3:	1	1	0

Gli altri bit della porta 2 non sono significativi.

Discussione

Nei precedenti esperimenti, avete controllato le tabelle della verità di diversi chip, tra i quali il 7400, 7402, 7408 e 7432. Ogni cambiamento di porta richiede un diverso chip e quindi alcuni cambiamenti nei collegamenti; invece nel microcomputer il cambiamento di una porta richiede solo il cambiamento di una o più istruzioni di "software". Usando il microcomputer potete fare delle sostituzioni col "software" cioè con istruzioni per computer; invece coi circuiti cablati sul breadboard fate sostituzioni con l'"hardware", cioè con circuiti integrati e fili. Ovviamente non userete mai il microcomputer MMD-1 per simulare una singola porta NAND o NOR, ma cosa fareste se, in un circuito digitale, vi servissero migliaia di porte? In un caso del genere potrebbe essere vantaggioso usare il microcomputer. Per esempio, un cambiamento nel comportamento di un circuito digitale, potrebbe essere eseguito cambiando semplicemente il software del microcomputer, invece di cambiare circuiti e collegamenti. È quindi chiaro che, una volta appurato che un circuito digitale può essere costruito con molte porte NAND e NOR, il microcomputer diventa un approccio valido solo per i circuiti che sono molto complessi.

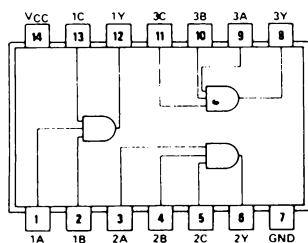
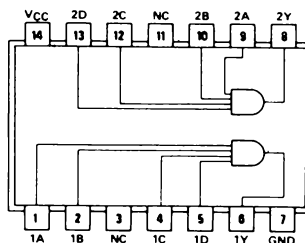
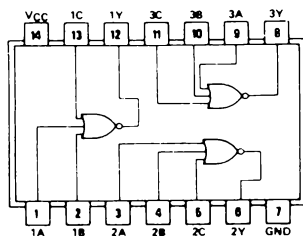
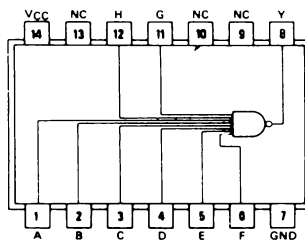
L'uso del microcomputer ha però una limitazione molto importante: la bassa velocità. Una delle caratteristiche tipiche dei "gate chips" è quella di essere molto veloci; infatti possono eseguire una operazione logica semplice in solo 10 ns, invece il microcomputer ha bisogno di un tempo, cento, mille volte più lungo. Quindi, se avete un'applicazione digitale che richiede alta velocità, non potreste usare uno dei microcomputer oggi in produzione.

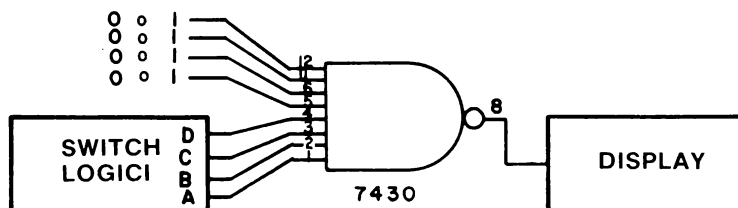
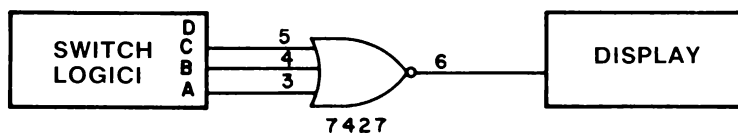
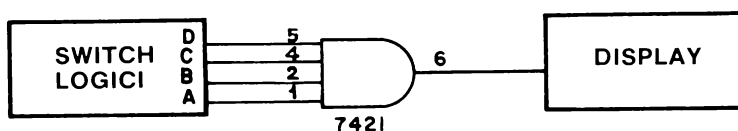
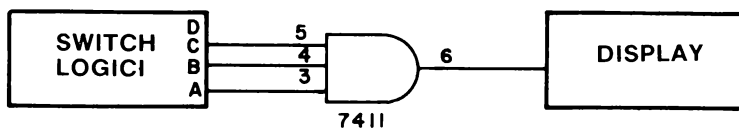
ESPERIMENTO N. 6

Scopo

Lo scopo di questo esperimento è di trovare la tabella della verità per ciascuno dei seguenti circuiti integrati della serie 7400:

- 7411, porta AND a 3-ingressi
- 74H21, porta AND a 4-ingressi
- 7427, porta NOR a 3-ingressi
- 7430, porta NAND a 3-ingressi

Configurazioni dei pin dei circuiti integrati

74H11

74H21

7427

7430

Schemi dei circuiti

Passo 1

I chip 74H11 e 7411 hanno le stesse caratteristiche operative, quindi vanno entrambi bene per eseguire questo esperimento. La stessa cosa vale anche per i chip 74H21 e 7421. La lettera "H" stà per "alta velocità" (high speed).

Passo 2

Collegate, nell'ordine dato, i chip al breadboard e determinate la tabella della verità di ciascuno di essi.

Cominciate col 7411, porta AND a 3-ingressi. La tabella della verità esatta è:

C	B	A	Q
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	1	0
1	1	0	0
1	1	1	1

Solo quando A, B e C sono all'1 logico, l'uscita Q è all'1 logico.

Passo 3

Scollegate l'alimentazione del breadboard e collegate il 7421 porta AND a 4-ingressi. Per questa porta dovreste trovare la seguente tabella della verità:

D	C	B	A	Q
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Solo quando A, B, C e D sono all'1 logico, l'uscita Q è all'1 logico.

Passo 4

Staccate l'alimentazione dal breadboard e collegate il 7427, porta NOR a 3-ingressi, che ha la seguente tabella della verità:

C	B	A	Q
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Solo quando A, B e C sono tutti allo 0 logico, l'uscita è all'1 logico. Evidentemente questa porta è diversa dalle due porte AND che avete appena provato.

Passo 5

Infine collegate il 7430, porta NAND a 4-ingressi. Di quale grandezza deve essere la tabella della verità per poter contenere tutte le possibili combinazioni degli ingressi?

Siccome ci sono otto ingressi, ci saranno $2^8 = 256$ diverse righe nella tabella della verità, quindi non vi chiederemo di costruirla.

Secondo voi, per quali condizioni dell'ingresso si ottiene lo stato logico dell'uscita, unico? Qual'è lo stato di questa uscita, lo 0 logico o l'1 logico?

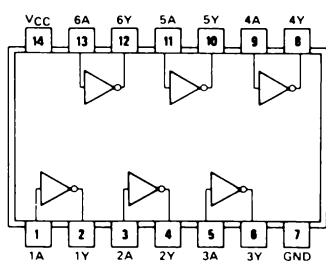
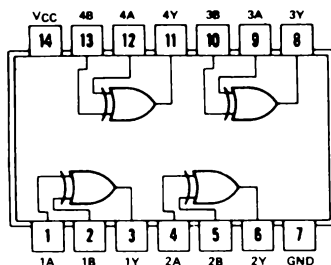
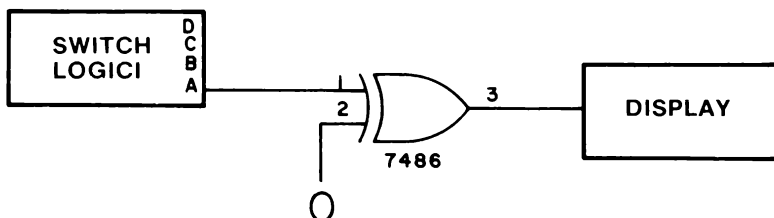
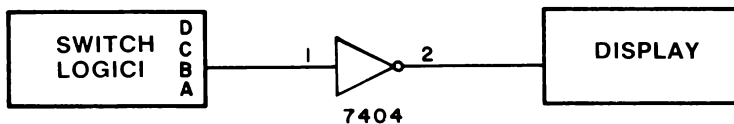
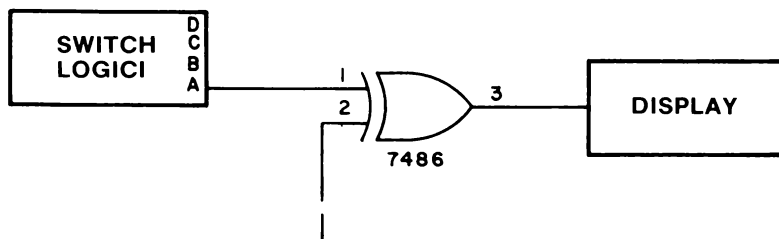
Vedrete lo 0 logico solo quando tutti e otto gli ingressi sono all'1 logico. La tabella della verità abbreviata può essere rappresentata nel modo seguente:

H	G	F	E	D	C	B	A	Q
[Tutte le altre combinazioni dell'ingresso]								1
1	1	1	1	1	1	1	1	0

ESPERIMENTO N. 7

Scopo

Lo scopo di questo esperimento è di mostrare l'uso come invertitore controllabile del 7486, porta OR-Esclusivo a 2-ingressi.

Configurazioni dei pin dei circuiti integrati

Schemi dei circuiti

Passo 1

Negli schemi che vi abbiamo appena dato, gli switch logici vengono usati per fornire un ingresso di dati digitali sia all'invertitore che alla porta OR-Esclusivo. Le domande, cui dovrete rispondere sperimentalmente sono:

- Che segnale, applicato al pin 2 del chip 7486, riesce ad invertire il dato in ingresso sul pin 1?
- Che segnale, applicato al pin 2 del chip 7486, lascia che il dato in ingresso al pin 1 passi direttamente al pin di uscita 3 senza venire invertito?

Collegate i circuiti degli schemi. Prima collegate il pin 2 del chip 7486 alla MASSA, che è allo 0 logico. Non dimenticate i collegamenti di alimentazione ai pin 7 e 14 sia nel chip 7404 che nel 7486.

Passo 2

Ridate alimentazione al breadboard e fate passare lo switch logico B dallo stato logico 0 allo stato logico 1. L'uscita che vedete sul display dovrebbe essere il complemento dell'ingresso applicato all'invertitore 7404. Detto in un altro modo, la tabella della verità per l'invertitore 7404 è la seguente:

A	Q
0	1
1	0

Passo 3

Adesso passate lo switch logico A dallo stato logico 0 allo stato logico 1. Dovreste vedere sul display che gli stati logici corrispondono alla seguente tabella della verità:

A	Q
0	0
1	1

In altre parole, quando il pin 2 del 7486 è allo 0 logico, il segnale in ingresso al pin 1 non viene invertito e passa attraverso la porta OR-Esclusivo.

Passo 4

Adesso collegate il pin 2 del chip 7486 al livello logico 1 e poi passate ancora una volta lo switch A dallo 0 logico all'1 logico. Dovreste adesso osservare che sul display i dati compaiono come se ci fosse un circuito invertitore

A	Q
0	1
1	0

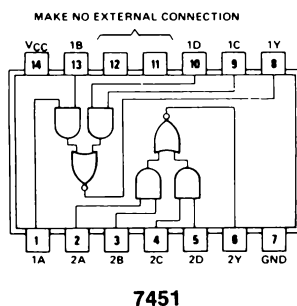
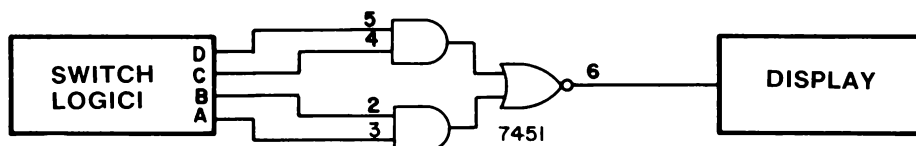
Allora, controllando lo stato logico del pin 2, potete far si che il chip 7486 inverta il segnale in ingresso o che lo lasci passare all'uscita senza invertirlo. Quindi avete costruito un semplice *invertitore controllabile* o *complementatore controllabile*.

ESPERIMENTO N. 8

Scopo

Lo scopo di questo esperimento è di determinare la tabella della verità per un 7451, porta AND-OR-INVERTITORE a 2-ingressi.

Configurazione dei pin del circuito integrato

7451

Schema del circuito

Passo 1

Collegate il circuito digitale mostrato nello schema e non dimenticate i pin di alimentazione del chip. Mettete tutti e quattro gli switch logici a 0 e alimentate il breadboard.

Passo 2

Variando la posizione dei quattro switch logici, eseguite tutti i controlli sperimentali necessari a completare la tabella della verità di un 7451, porta AND-OR-INVERTITORE 2-wide, 2-input, che riportiamo di seguito:

D	C	B	A	Q
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Quando abbiamo fatto questo esperimento, abbiamo ottenuto i seguenti risultati:

D	C	B	A	Q
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

DOMANDE RIEPILOGATIVE

Le seguenti domande vi aiuteranno a ripassare i circuiti integrati che abbiamo discusso in questo capitolo.

1. Quando si dice "circuito integrato", cosa si intende per "integrato"?

2. Come localizzate il pin 1 di un circuito integrato?

3. Quali collegamenti di alimentazione bisogna fare ad un circuito integrato della serie 7400?

4. Quante porte ci sono in ciascuno dei seguenti chip?

   a. 7400

   b. 7427

   c. 7402

   d. 7486

   e. 7420

   f. 7410

   g. 7421

   h. 7451

5. Qual'è lo stato logico di un ingresso non collegato di un qualsiasi circuito integrato della serie 7400.

6. Quale sottofamiglia TTL assorbe meno tensione?

7. Qual'è la differenza tra un fan-in e fan-out?

8. Qui sotto vi diamo un elenco di numeri di identificazione che di solito si trovano sul lato superiore dei circuiti integrati. Trovate, in ognuno di essi, il numero del chip.

   a. SW 74181N 7307

   b. SW 74150N 7305

   c. Sprague 7241 US 7430A

   d. NS 330 DM7472N

   e. DM/SN7401N NS8001N 137

   f. S7237 N7411A

   g. NS 334 DM7447AN

   h. S6926 N7402A

RISPOSTE

1. Il verbo "integrare" significa "mettere insieme, partendo da parti separate, in un unico; unire", quindi "integrato" si riferisce al fatto che gli elementi individuali del circuito, come resistori, transistor e diodi, vengono messi insieme per formare un elemento unico.

2. Bisogna cercare un segno di identificazione su un bordo del chip. Questo segno può essere una tacca semicircolare, un punto dipinto, un incavo sul contenitore plastico, una tacca di qualsiasi forma etc.

3. Solo le tensioni di +5V e MASSA.

4. a. quattro
   b. tre
   c. quattro
   d. quattro
   e. due
   f. tre
   g. due
   h. due

5. All'1 logico in ogni caso

6. TTL a bassa potenza, seguita dallo Schottky TTL a bassa potenza.

7. Il fan-in si riferisce all'ammontare di corrente che un ingresso ad un chip TTL fornisce quando è collegato all'uscita di un altro chip TTL. La grandezza di questa corrente è 1,6 mA. Il fan-out si riferisce all'ammontare di corrente che un'uscita da un chip TTL può pilotare o assorbire. La grandezza di questa corrente è 16 mA.

8. a. 74181
   b. 74150
   c. 7430
   d. 7427
   e. 7401
   f. 7411
   g. 7447
   h. 7402

CAPITOLO 11

FLIP-FLOP E LATCH

INTRODUZIONE

Questo capitolo vi introdurrà a quella parte di logica in cui vengono usati i segnali di clock (clocked logic) ed a due tipi molto usati di flip-flop, il 7474 tipo D positive-edge-triggered e il latch 7475. Negli esperimenti, servirà un circuito single-step per il microcomputer MMD-1, o per ogni altro sistema microcomputer basato sull'8080A e che usi il chip 8224.

OBIETTIVI

Alla fine di questo capitolo sarete in grado di:

- Dare la definizione di elemento bistabile, flip-flop, ingresso sincrono, ingresso asincrono, flip-flop tipo-D, latch, preset, clear e forma d'onda digitale.
- Distinguere tra fronte positivo e fronte negativo di un impulso di clock.
- Spiegare come un latch, composto di due porte NAND a 2 ingressi, possa essere usato per fornire una logica antirimbalzo ad uno switch meccanico SPDT.
- Descrivere le caratteristiche di funzionamento del 7474, flip-flop positive-edge-triggered (flip-flop sensibile al fronte positivo).
- Descrivere le caratteristiche di funzionamento del latch 7475.
- Spiegare come funzionano gli ingressi preset e clear di un tipico flip-flop.
- Descrivere i tre tipi di azioni di clock in un flip-flop.
- Disegnare uno schema per un circuito single-step che possa essere usato con un microcomputer basato sull'8080A e che impieghi il chip di controllo 8224.

CLOCKED LOGIC

Riportiamo tra virgolette quanto ha detto Lancaster [D.E. Lancaster, "TTL Cookbook", Howard W. Sams & Co., Inc., Indianapolis, 1974] "I sistemi a logica *sottoposta a clock, sincrona,* detta anche *passo passo (step by step)* sono la maggior parte dei sofisticati circuiti digitali attuali, specialmente a livello di package o di dispositivo. Invece di fornire l'uscita non appena siano cambiate le condizioni dell'ingresso i package clocked logic, attendono sino a che non arrivi un impulso di clock o un comando di sistema; solo allora rispondono fornendo un'uscita".

"Ci sono molti vantaggi ad usare la clocked logic. Il primo è che un transitorio non controllato non può percorrere a piacimento il circuito, e quindi i cambiamenti possono prodursi solo contemporaneamente e in modo ordinato. Questo modo di operare "un passo alla volta" rende i dispositivi simili a dei registri di shift e a dei contatori molto complessi. Il secondo vantaggio della clocked logic è che, nel sistema, tutto accade più o meno allo stesso istante e questo elimina o quanto meno minimizza, la race condition, i glitch, e l'accavallarsi di sequenze temporali".

"Le pietre miliari della clocked logic sono i flip-flop JK e i flip-flop tipo D. . . . .".

ELEMENTI DI MEMORIA: FLIP-FLOP

Un elemento di memoria è, in genere, un qualsiasi dispositivo che può immagazzinare bit allo 0 o 1 logico in modo tale che un singolo bit o un gruppo di bit sia accessibile e riutilizzabile. La più comune forma di memoria dell'elettronica digitale è *l'elemento bistabile,* più conosciuto come *flip-flop.* Una buona definizione di questi due elementi è stata data da H.V. Malnustadt e C. G. Enke in "Digital Electronics for Scientists" [W.A. Benjamin, Inc., New York, 1967]:

Elemento bistabile — Altro nome di flip-flop. Circuito nel quale l'uscita ha due stati stabili (stati di uscita 0 o 1) e può essere pilotato a uno di questi due stati dai segnali di ingresso, ma che rimane permanente in questo stato anche dopo che il segnale di ingresso è cessato. Questo fatto differenzia un elemento bistabile da una porta, che possiede due stati di uscita ma che ha bisogno di un ingresso continuo per restare nello stato in cui si trova. La caratteristica di avere due stati stabili lo differenzia dai monostabili, che ritornano sempre ad uno stato specifico, e dagli elementi instabili che cambiano sempre da uno stato all'altro.

Flip-flop — Circuito con due stati stabili che ha la capacità di passare da uno stato all'altro mediante un segnale di controllo, e che rimane in questo stato dopo che il segnale è cessato.

In altre parole, flip-flop è un termine che si adatta a tutti i bistabili, o dispositivi a due stati, e che quindi comprende dispositivi di diversi tipi:

- elettronici, come i flip-flop a semiconduttore tipo D, flip-flop JK, flip-flop RS, flip-flop RST, e flip-flop T;
- magnetici, come nuclei magnetici, dischi magnetici, nastri magnetici, cavi magnetici e bolle magnetiche;
- chimici, come quei dispositivi basati sui fenomeni elettrochimici;
- meccanici, come pull-chain light switch;
- fluidi, come i flip-flop fluidi.

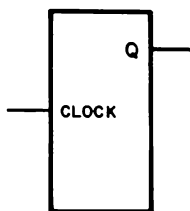
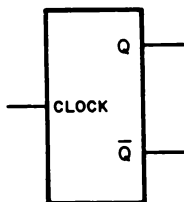
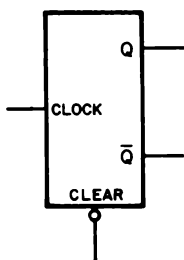
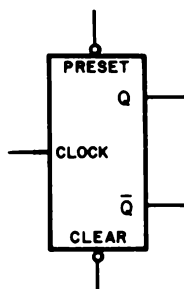
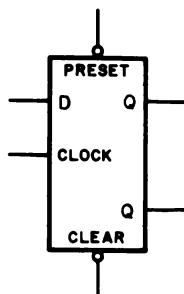
In questo Bugbook, farete conoscenza solo con i flip-flop a semiconduttore, principalmente di tipo D, che sono conosciuti anche come *latch.*

Vorremmo sottolineare adesso, una delle differenze tra porte e flip-flop:

> *Una porta, per rimanere in un determinato stato, ha bisogno di avere un ingresso continuo, cosa che non è necessaria per i flip-flop.*

Gli elementi di memoria permettono di immagazzinare delle informazioni digitali per un uso successivo; una possibilità di questo tipo permette, oltre ad altre cose, di costruire delle macchine elettroniche digitali, dette *microcomputer, minicomputer,* o *computer.* Indipendentemente dalle loro dimensioni, tutti i computer usano da qualche migliaio a qualche milione di elementi di memoria e tutti i computer funzionano immagazzinando e trattando solo due stati logici: lo 0 logico e l'1 logico. Siccome i computer esistenti dipendono in gran parte dagli elementi di memoria magnetici quali nuclei e dischi magnetici, molte persone, che lavorano nell'industria elettronica, prevedono che i dispositivi a semiconduttore di memoria giocheranno un ruolo sempre più importante nella memorizzazione dei computer delle informazioni digitali. Questo è già vero oggi per i minicomputer e per i microcomputer.

ALCUNI SEMPLICI FLIP-FLOP

Esistono diversi tipi di flip-flop, ma il più semplice di essi ha un solo ingresso di clock e una sola uscita, Q;

Un flip-flop di questo tipo può avere anche *un'uscita complementare,* $\overline{Q}$, che si trova sempre allo stato logico opposto a quello di Q;

un ingresso di clear, che permette di mettere Q allo 0 logico;

un ingresso di preset, che permette di mettere Q all'1 logico;

e un ingresso di dati, D, che viene messo all'uscita Q quando arriva l'impulso di clock. L'azione condizionata dal clock può avvenire in vari momenti dell'impulso di clock, come vedremo più tardi.

Tra gli altri tipi di flip-flop ricordiamo: flip-flop JK, latch RS (conosciuto anche come flip-flop RS), clocked latch RS e flip-flop T. Per il momento non è necessario che ne sappiate di più.

Riportiamo alcune definizioni in parte riprodotte per concessione di Malnustadt ed Enke

Flip-flop D — D sta per DATI. Un flip-flop la cui uscita è funzione dell'ingresso presente all'ultimo impulso di clock; per esempio, se in ingresso c'è l'1 logico, l'uscita, dopo il prossimo impulso di clock, sarà all'1 logico.

Clock — Un generatore di impulsi che controlla la temporizzazione dei dispositivi logici sottoposti a clocked logic e che regola la velocità alla quale questi dispositivi devono operare. Serve a sincronizzare tutte le operazioni che si svolgono in un sistema digitale. Un clock può essere ogni dispositivo digitale che genera uno ò più impulsi di clock.

Ingresso di clock — Quel terminale di un flip-flop la cui condizione o cambiamento di condizione controlla l'accettazione dei dati nel flip-flop per mezzo di ingressi sincroni, e che quindi controlla lo stato dell'uscita del flip-flop. Il segnale di clock ha due funzioni: (1) permette ai segnali dei dati di entrare nel flip-flop; (2) dopo l'ingresso, dirige il flip-flop affinchè cambi stato in base all'ingresso avuto.

Sincrona — Operazione di un sistema clocked logic compiuta con l'aiuto di un generatore di impulsi di clock. Tutte le azioni vengono in sincronia con il clock.

Ingresso asincroni — Quei pin di ingresso di un flip-flop che possono influenzare lo stato di uscita del flip-flop indipendentemente dal clock. Sono detti preset e reset o clear.

Ingresso sincroni — Quei pin di un flip-flop attraverso i quali i dati possono entrare, ma solo sotto comando del clock. Questi ingressi hanno sempre il controllo diretto dell'uscita come accade per quelli di una porta, ma l'hanno solo quando il clock lo permette e lo comanda. Sono detti anche ingressi JK o D.

Preset — Ingresso asincroco che viene usato per controllare lo stato logico dell'uscita Q. I segnali che entrano da questo ingresso mettono l'uscita Q all'1 logico e non possono mai mettere l'uscita Q allo 0 logico.

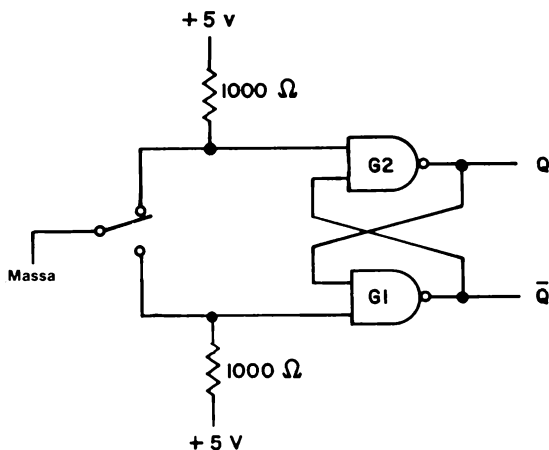
Clear — Ingresso asincrono che viene usato per controllare lo stato logico dell'uscita Q. I segnali che entrano da questo ingresso mettono l'uscita Q allo 0 logico e non possono mettere l'uscita Q all'1 logico.

Latch — Elemento semplice per memorizzazione logica. Un loop in feedback usato in un circuito digitale simmetrico (come un flip-flop) per ritenere uno stato logico.

Il latch viene usato nei circuiti digitali principalmente come elemento semplice di memoria.

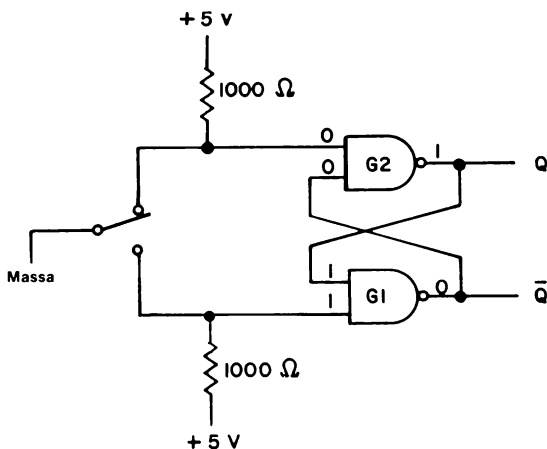
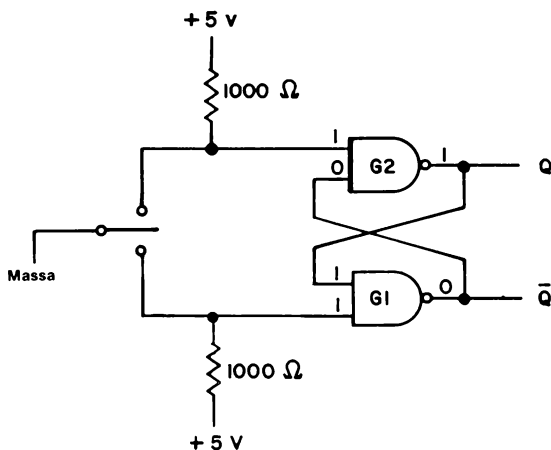
UN LATCH SEMPLICE

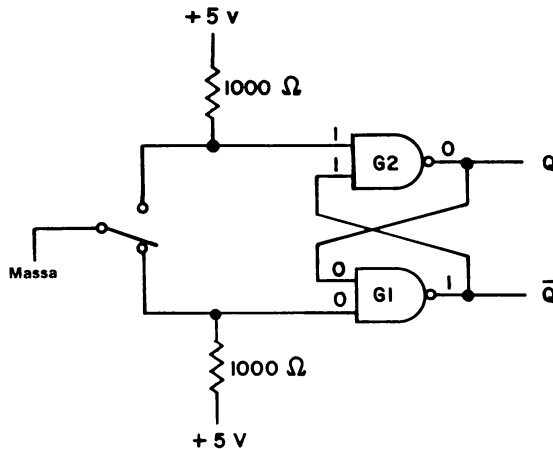
Discuteremo adesso delle proprietà di un latch che è un bistabile molto semplice e che può essere costruito con un paio di 7400, porte NAND positive a 2 ingressi, un paio di resistori da 1000Ω o uno switch SPDT (single-pole double throw), oppure un filo collegato a massa. Un elemento bistabile è un dispositivo che è capace di assumere stabilmente uno dei due stati logici. Questo circuito è uno di quelli usati nell'Outboard LR-6, doppio generatore d'impulsi, o nell'Outboard LR-25, breadboarding station, per dotare di logica antirimbalzo lo switch meccanico SPDT (guardate la Figura A-7 dell'Appendice 3).

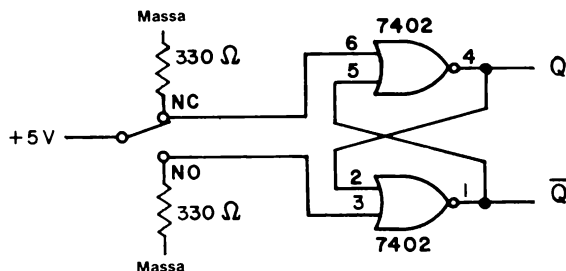
Qui sotto riportiamo lo schema di un latch bistabile,

B	A	Q
0	0	1
0	1	1
1	0	1
1	1	0

Ricordate che l'uscita di una porta NAND a 2 ingressi è allo 0 logico quando entrambi gli ingressi sono all'1 logico. Adesso possiamo assegnare i seguenti stati logici ad ogni singolo ingresso e uscita dello schema (guardate lo schema seguente). Per ottenere questi risultati, usiamo il seguente ragionamento:

> L'ingresso in alto della porta G2 è allo 0 logico perchè è messo a massa, quindi *l'uscita della porta G2 deve essere all'1 logico.* Ma siccome l'uscita della porta G2 è collegata ad un ingresso della porta G1, questo ingresso sarà all'1 logico; ma anche l'altro ingresso della porta G1 è all'1 logico perchè è "tenuto su" dai +5V per mezzo del resistore da 1000Ω. I due ingressi della porta G1 all'1 logico producono un'uscita allo 0 logico che viene collegata all'altro ingresso della porta G2.

Se non si sposta lo switch non si avrà alcun mutamento dell'uscita del latch bistabile che rimarrà allo stato indicato per un tempo indefinito.

Adesso spostiamo lo switch dal contatto superiore senza metterlo in contatto col filo di collegamento più in basso,

Vi chiediamo: si verifica qualche cambiamento alle uscite Q e $\overline{Q}$? La risposta, come illustrato nello schema che precede, è no! Lo switch è tra i due contatti; è già staccato dal contatto superiore ma non ha ancora toccato quello inferiore. L'unico cambiamento, come potete osservare, è avvenuto nell'ingresso superiore della porta G2 che adesso è all'1 logico, ma questo cambiamento non è tale da portare l'uscita al suo stato unico (lo 0 logico). Si dice che l'uscita Q è "latched" (bloccata) allo stato logico 1.

Come ultimo passo di questa discussione, facciamo toccare allo switch il contatto inferiore, come mostrato dal prossimo schema,

Adesso cambiano alcuni stati logici. L'ingresso più in basso della porta G1 è messo a massa e quindi allo stato logico 0. *L'uscita della porta G1 viene spinta allo stato logico 1.* Questa stessa uscita serve come ingresso della porta G2, quindi, siccome tutti e due gli ingressi di questa porta sono adesso all'1 logico, si ha lo stato unico dell'uscita, cioè lo 0 logico. L'uscita della porta G2 è in collegamento con l'ingresso in alto della porta G1. Adesso il circuito è bloccato allo stato logico 0. Se il contatto dello switch "rimbalza" e si stacca dal collegamento inferiore, *ma non tocca quello superiore,* il circuito resta comunque bloccato allo 0 logico. Questo è il metodo con cui si fornisce a uno switch meccanico una logica antirimbalzo.

Per dotare uno switch meccanico di logica antirimbalzo si può usare anche una coppia di 7402, porte NOR a 2 ingressi,

però vi sconsigliamo di usare un circuito di questo tipo perchè assorbe il triplo della corrente che consuma quello fatto con le porte NAND e inoltre, è più difficile "abbassare" che "alzare", per mezzo di un resistore, un livello logico.

Ogni funzione logica, del semplice flip-flop al computer, può essere prodotta con una appropriata combinazione di porte NAND o NOR.

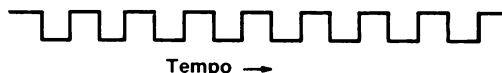
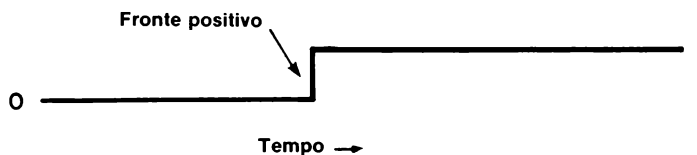
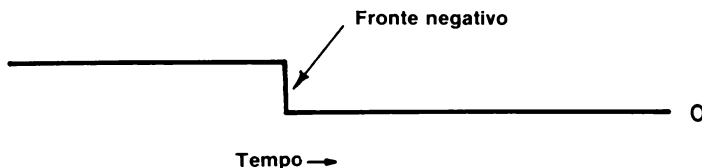
Una discussione sui flip-flop più approfondita è comunque aldilà dello scopo di questo Bugbook.

FRONTI POSITIVI E NEGATIVI

Possiamo definire il termine, *forma d'onda digitale,* nel modo seguente:

Forma d'onda digitale — Rappresentazione grafica di un segnale digitale che illustra la variazione di uno stato logico in funzione del tempo. Questo tipo di rappresentazione è conosciuto come *diagramma temporale.*

La ragione principale per cui si usano le forme d'onda digitali è *permettere di mostrare le condizioni logiche che esistono, in un particolare istante, in un circuito digitale complesso.* In realtà un circuito digitale non ha bisogno di essere complesso, quindi l'uso della forma d'onda digitale è essenziale solo per i circuiti complessi. Per un circuito logico semplice, può essere necessario illustrare le forme d'onda solo in uno, o pochi altri, punti del circuito. Per i circuiti complessi, sono invece necessarie dieci o quindici forme d'onda per comprendere cosa succede.

Una delle forme d'onda più comuni è quella del *treno di impulsi di clock:*

In ogni forma d'onda, il tempo va da sinistra a destra, lo stato inferiore, o linea base, è di solito lo stato logico 0 e la parte superiore dell'impulso è lo stato logico 1, cioè i +2,5V o più. Noi supponiamo che il cambiamento dallo stato logico 0 allo stato logico 1, che è detto fronte positivo, avvenga molto rapidamente (in alcuni nanosecondi o meno) e che sia quindi praticamente istantaneo. Nella forma d'onda, rappresentiamo questo cambiamento con una linea verticale come mostrato di seguito:

Rappresentiamo, invece, un cambiamento dall'1 logico allo 0 logico con una linea verticale detta *fronte negativo,*

Supponiamo anche in questo caso che la transizione avvenga istantaneamente, mentre in realtà non lo è. Esiste sia un *tempo di salita* che un *tempo di discesa,* le cui definizioni sono le seguenti:

Tempo di salita — Tempo necessario affinchè il fronte positivo di un impulso passi dal 10% al 90% del suo valore finale. È proporzionale alla costante di tempo ed è misura della pendenza del fronte d'onda. In elettronica digitale è la quantità di tempo richiesta perchè la tensione in uscita di un circuito digitale cambi dal livello di tensione più basso (0 logico) al livello di tensione più alto (1 logico)[2].

Tempo di discesa — Tempo necessario affinchè un fronte negativo di un impulso passi dal 90% al 10% del suo valore iniziale. In elettronica digitale è la quantità di tempo richiesta perchè la tensione di uscita di un circuito digitale cambi dal livello di tensione più alto (1 logico) al livello di tensione più basso (0 logico)[2].

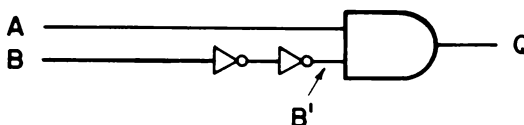
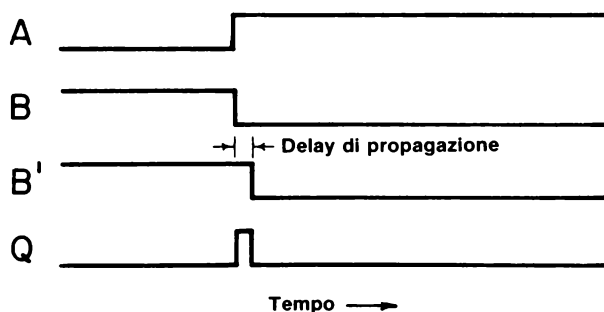
Quando si parla di tempo di salita e tempo di discesa è naturale definire anche il *delay di propagazione* (o tempo totale di ritardo di propagazione):

Tempo totale di ritardo di propagazione — È la conseguenza di quattro ritardi tipici dei circuiti - memorizzazione, salita, discesa, cambiamento di stato - ed è il tempo che intercorre da quando il segnale in ingresso passa il livello di soglia di tensione a quando la tensione di risposta in uscita passa il corrispondente livello di soglia di tensione.

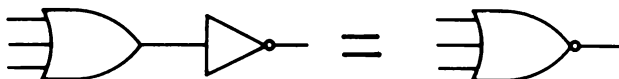
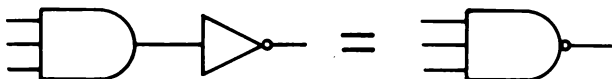
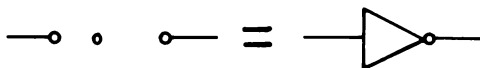
I delay di propagazione delle porte sono stati elencati, divisi per tipi di porte TTL, nel capitolo 10. Il delay di propagazione di una porta TTL normale è di circa 10 ns = 10^{-8} s. Una porta TTL a bassa potenza ha un delay di propagazione maggiore di 33 ns, mentre un chip Schottky TTL ha un delay di propagazione di solo 3 ns. Quando si lavora a frequenze digitali molto alte con dei circuiti logici molto complessi, il concetto di delay di propagazione diventa molto importante. Può essere importante anche quando si fa del breadboarding con chip digitali e si eseguono i test a bassa frequenza, ma negli esperimenti di questa serie di capitoli non viene preso in considerazione. Un microcomputer basato sull'8080A opera a frequenze comprese tra 500 kHz e 2 MHz che non sono frequenze molto alte e quindi non si avranno mai problemi di delay di propagazione con i chip della serie 7400 a meno che il progetto non sia in race condition.

Il termine race può essere definito nel modo seguente:

Race — Condizione che si verifica quando il cambiamento di stato di un sistema richiede il cambiamento di due o più stati semplici. Se lo stato finale è influenzato anche da una variabile che può cambiare stato prima che le altre variabili abbiano cambiato il loro, ci si trova in un race condition. È anche la condizione che si verifica quando c'è un segnale che si propaga attraverso due o più elementi di memoria durante lo stesso periodo di clock.

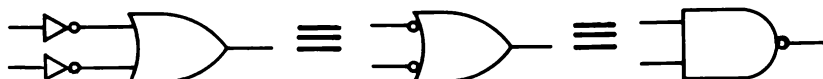
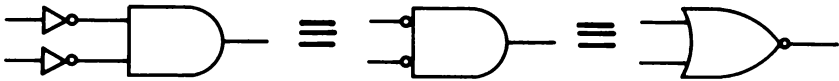
Si ha una race condition quando due segnali che dipendono dal tempo si "inseguono" producendo con il loro inseguimento ùna azione od un gruppo di azioni. La race condition sono di solito causate da una progettazione poco curata o dai delay di propagazione. Nel seguente esempio di verifica una race condition a causa dei delay di propagazione dei due invertitori, se si suppone che A e B cambino stato nello stesso istante,

In questo caso, si vorrebbe che l'uscita Q restasse allo 0 logico perchè A e B cambiano stato contemporaneamente e quindi lo stato unico, A=B=1, non viene mai raggiunto. Come mostrato nel seguente diagramma temporale:

il tempo di ritardo di propagazione degli invertitori produce un ritardo e dà luogo al "glitch" di Q, infatti la condizione logica B=0 ha bisogno di tempo per attraversare gli invertitori.

SIMBOLOGIA RELATIVA ALL'INVERSIONE

Avete già visto nei capitoli precedenti che i piccoli cerchietti che si trovano sull'uscita delle porte AND e OR significano *inversione*,

Questi cerchietti appaiono frequentemente negli schemi dei dispositivi digitali e dei circuiti, e quindi è buona norma discutere dei loro possibili significati, ma prima di continuare, dobbiamo dare delle definizioni:

Disabilitare, inibire	Impedire ad un segnale digitale di entrare o passare in un dispositivo digitale o in un circuito.
Abilitare, fornire un segnale di abilitazione (strobe), fornire un trigger	Permettere ad un segnale digitale di entrare o passare in un dispositivo digitale o in un circuito.

Quando viene applicato ad un dispositivo clocked logic, il termine dare un clock (to clock), è frequentemente usato per intendere il passaggio di un segnale digitale all'interno di un circuito o il cambiamento di stato di un elemento bistabile. Noi useremo indifferentemente in questo testo, i due termini.

I cerchietti di inversione possono essere applicati sia alle uscite (come nelle porte NAND e NOR) sia agli ingressi delle porte. Per esempio, il teorema di De Morgan che abbiamo visto nel capitolo 8, può essere rappresentato nei seguenti modi:

dove naturalmente il cerchietto di inversione stà a significare un invertitore.

I possibili significati dei cerchietti di inversione agli ingressi o alle uscite dei dispositivi digitali, possono essere riassunti nel seguente modo. *State attenti che queste convenzioni vengono di solito usate per i dispositivi bistabili o clocked logic e difficilmente vengono usati per le porte!*

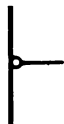
Tabella 5 Possibili significati dei cerchietti di inversione all'ingresso o all'uscita dei dispositivi digitali o dei circuiti, porte escluse.

INGRESSI

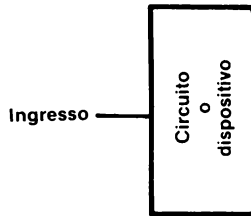
L'1 logico abilita il dispositivo o il circuito
Lo 0 logico disabilita il dispositivo o il circuito
Il fronte positivo abilita, fornisce uno strobe, fornisce un trigger o sottopone a clock il dispositivo o il circuito
L'impulso di clock positivo fornisce uno strobe o sottopone a clock il dispositivo o il circuito
I dati si presentano senza essere invertiti

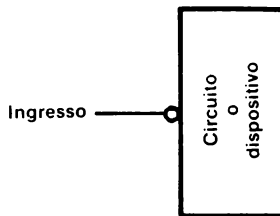
Lo 0 logico abilita il dispositivo o il circuito
L'1 logico disabilita il dispositivo o il circuito
Il fronte negativo abilita, fornisce uno strobe, un trigger o sottopone a clock il dispositivo o il circuito.
I dati si presentano invertiti

USCITA

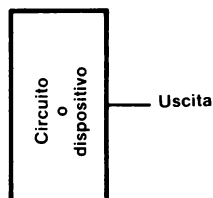
L'uscita dal dispositivo o dal circuito non è invertita
L'uscita dal dispositivo o dal circuito è un impulso di clock positivo

L'uscita dal dispositivo o dal circuito è invertita
L'uscita dal dispositivo o dal circuito è un impulso di clock negativo

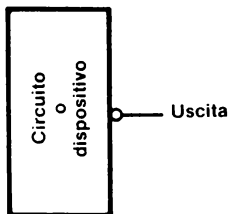
Per gli ingressi dei dispositivi o circuiti a clocked logic si applicano le seguenti convenzioni:

Se non c'è nessun cerchietto all'intersezione della linea che rappresenta l'ingresso e lo schema a blocchi:

allora, un impulso di clock positivo o un fronte positivo, abilita, fornisce uno strobe, sottopone a clock o trigger, il dispositivo o il circuito affinchè esegua una azione.

Se c'è un cerchietto all'intersezione della linea che rappresenta l'ingresso e lo schema a blocchi,

allora, un impulso di clock negativo o un fronte negativo, abilita, fornisce uno strobe, sottopone a clock o trigger il dispositivo o il circuito affinchè esegua una azione.

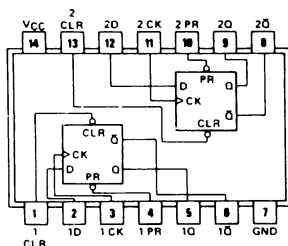
Per le uscite dai dispositivi digitali e dai circuiti, si applica la seguente convenzione:

Se non c'è nessun cerchietto all'intersezione della linea che rappresenta l'uscita e lo schema a blocchi:

Allora l'uscita del circuito o del dispositivo non è invertita ed è considerato unico lo stato logico 1.

Se c'è un cerchietto all'intersezione della linea che rappresenta l'uscita e lo schema a blocchi:

Allora l'uscita del circuito o del dispositivo è invertita ed è considerato unico lo stato logico 0.

Generalmente gli schemi dei chip e dei circuiti forniti dai costruttori sono affidabili per quanto riguarda la presenza o l'assenza dei cerchi di inversione, nei trattati invece, è facile che siano trascurati. Spesso ci vuole troppo tempo per disegnare ad esempio sedici cerchietti di inversione sulle uscite del chip decodificatore 74154. Se ci sono i cerchietti di inversione vuol dire che esiste un motivo importante, quindi, quando li vedete, state attenti.

7474, D TYPE POSITIVE EDGE TRIGGERED FLIP-FLOP

Il nostro flip-flop preferito, che in realtà è un latch, è il *7474, flip-flop tipo D positive-edge-triggered con preset e clear,* la cui configurazione dei pin è la seguente:

Ogni chip contiene due latch di tipo D, come si può chiaramente vedere dallo schema qui sopra.

La tabella della verità di questo chip, come quella di tutti i flip-flop della serie 7400, è riportata nel "The TTL Data-Book for Design Engineers" della TEXAS INSTRUMENTS INCORPORATED e nel Bugbook II. Per capire questo tipo di tabella della verità, bisogna conoscere le seguenti definizioni:

- H sta per 1 logico
- L sta per 0 logico
- Q_0 è il livello di Q prima che fossero stabilite le condizioni di ingresso indicate
- $\overline{Q}_0$ è il complemento di Q_0
- * stà a significare che la configurazione non è stabile; cioè non continua a sussistere quando gli ingressi preset e clear ritornano al loro stato di inattività (1 logico)
- ↑ stà a significare un ingresso di clock positive-edge-triggered
- ↓ stà a significare un ingresso di clock negative-edge-triggered
- X stà a significare che lo stato di questo ingresso è non significativo; potrebbe essere sia lo 0 che l'1 logico e non avrebbe alcuna influenza sullo stato logico dell'uscita anche dopo l'attivazione del clock.
- TOGGLE stà a significare che ogni uscita, Q e $\overline{Q}$, si trasforma nel proprio complemento dopo ogni attivazione del clock.

Con queste definizioni potete adesso capire le tabelle della verità che fornisce la Texas Instruments Incorporated per il flip-flop di Tipo-D, 7474,

74 DUAL D-TYPE POSITIVE-EDGE-TRIGGERED FLIP-FLOPS

FUNCTION TABLE

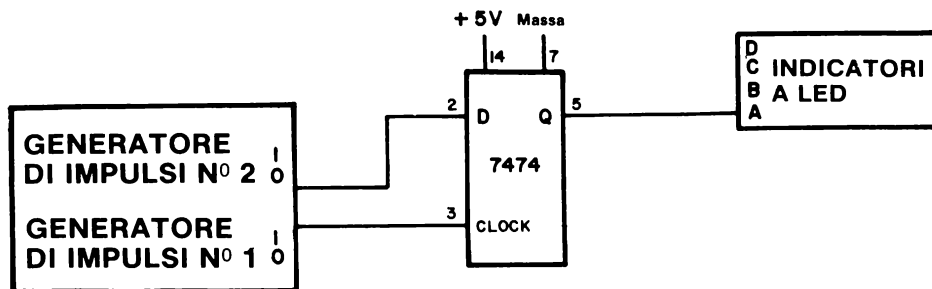
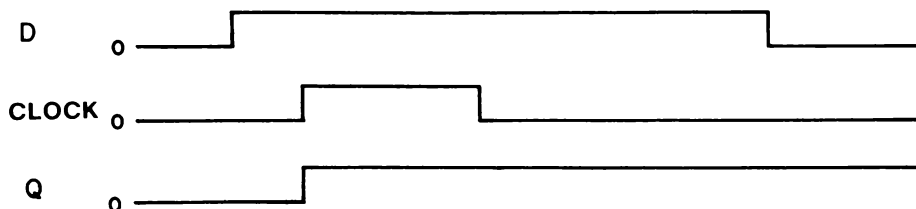
INPUTS				OUTPUTS	
PRESET	CLEAR	CLOCK	D	Q	$\overline{Q}$
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H*	H*
H	H	↑	H	H	L
H	H	↑	L	L	H
H	H	L	X	Q_0	$\overline{Q}_0$

Potremmo anche riscrivere questa tabella della verità in termini di stati logici 0 e 1:

Ingressi				Uscite	
PRESET	CLEAR	CLOCK	D	Q	$\overline{Q}$
0	1	X	X	1	0
1	0	X	X	0	1
0	0	X	X	*	*
1	1	↑	1	1	0
1	1	↑	0	0	1
1	1	0	X	Q_0	$\overline{Q}_0$

Preset e CLEAR sono gli ingressi asincroni e quando sono nello stesso stato logico 0 prevaricano gli ingressi CLOCK e D. Sia il PRESET che il CLEAR possono essere contemporaneamente allo stato 0 logico, ma, in tal caso, non è possibile prevedere il risultato. È una condizione che bisogna evitare già in progettazione.

UN SEMPLICE CIRCUITO 7474

Uno dei primi esperimenti che farete col latch di tipo-D 7474, sarà di osservare come l'ingresso di CLOCK e l'ingresso D influenzano l'uscita Q del latch,

L'uscita Q è dipendente dallo stato logico dell'ingresso D *quando il fronte positivo di ogni impulso di clock giunge all'ingresso CLOCK.* Per esempio, considerate il seguente diagramma temporale:

Osservate quando l'uscita Q passa allo stato logico 1. Lo fa quando giunge il fronte positivo del segnale di CLOCK. Una volta che l'uscita è bloccata all'1 logico, rimane in questo stato fino a quando D passa allo 0 logico e arriva un altro fronte positivo. Quando studiate i diagrammi temporali, ricordatevi che il tempo va da sinistra a destra, quindi i primi eventi sono quelli più a sinistra. Di solito non vengono riportati i delay di propagazione.

INGRESSI PRESET E CLEAR

Agli ingressi PRESET e CLEAR di un flip-flop, si applicano le seguenti regole:

> *Gli ingressi PRESET e CLEAR non devono essere contemporaneamente abilitati, si avrebbe in uscita uno stato non prevedibile.*
>
> *Quando l'ingresso PRESET è abilitato, prevarica tutti gli ingressi sincroni, quali CLOCK, D, J e K, e quindi manda l'uscita Q del flip-flop all'1 logico.*
>
> *Quando l'ingresso CLEAR è abilitato, prevarica tutti gli ingressi sincroni, quali CLOCK, D, J e K, e quindi manda l'uscita Q del flip-flop allo 0 logico.*

Gli ingressi sincroni di un flip-flop possono essere abilitati solo disabilitando l'ingresso PRESET e l'ingresso CLEAR.

Molti, se non tutti i flip-flop della serie 7400, hanno i cerchietti di inversione sugli ingressi PRESET e CLEAR, quindi possiamo aggiungere anche questa regola:

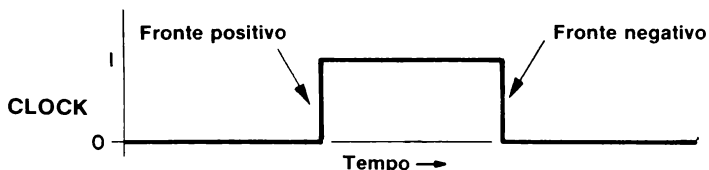
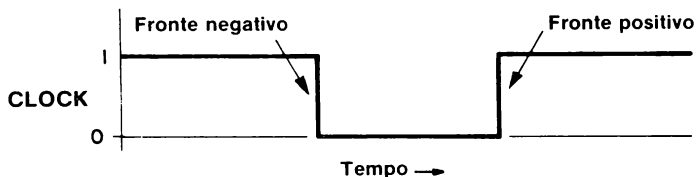
Uno stato logico 0 applicato all'ingresso PRESET metterà l'uscita del flip-flop all'1 logico; uno stato logico 0 applicato all'ingresso CLEAR metterà l'uscita del flip-flop allo 0 logico; gli stati logici 0 non devono essere contemporaneamente applicati agli ingressi PRESET e CLEAR di un flip-flop.

Il simbolo dell'impulso negativo di clock

viene di solito usato per indicare che su un dato flip-flop esistono sia l'ingresso PRESET che l'ingresso CLEAR. I seguenti flip-flop hanno il cerchietto di inversione agli ingressi PRESET e CLEAR: 7470, 74H71, 7472, 7473, 7474, 7476, 7478, 74101, 74102, 74103, 74106, 74107, 74108, 74109, 74110, 74111, 74112, 74113, e 74114.

FLIP-FLOP EDGE-(LEVEL)-TRIGGERED

Tutti i flip-flop sono dei dispositivi a clocked logic. Invece di fornire un'uscita subito dopo un cambiamento delle condizioni di ingresso - come è caso di tutte le porte - i flip-flop attendono l'arrivo di un impulso di clock e solo allora rispondono fornendo una nuova uscita. Il termine, triggered (da trigger=grilletto) è sinonimo del termine clocked e stà a significare l'applicazione di un singolo impulso di clock all'ingresso di CLOCK del flip-flop.

Gli impulsi di clock possono essere positivi o negativi. Un impulso di clock positivo è un ciclo logico completo dallo 0 logico all'1 logico e poi ancora allo 0 logico:

Un impulso di clock negativo è un ciclo logico completo dall'1 logico allo 0 logico e poi ancora all'1 logico:

I tre modi di azione del clock, possono essere così descritti:

- *Flip-flop Positive-edge-triggered* (innescato da un fronte positivo)

  Il trasferimento di informazioni dall'ingresso (dagli ingressi) sincrono (i) all'uscita dei flip-flop avviene quando giunge il fronte positivo dell'impulso di clock.

- *Flip-flop Negative-edge-triggered* (innescato da un fronte negativo)

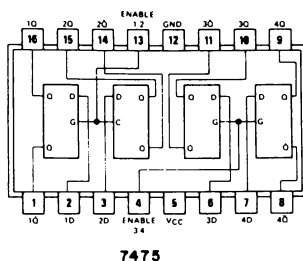
  Il trasferimento di informazioni dall'ingresso (dagli ingressi) sincrono (i) all'uscita del flip-flop avviene quando giunge il fronte negativo dell'impulso di clock.

- *Flip-flop Level-triggered* (innescato dal livello)

  Quando lo stato dell'ingresso di CLOCK passa allo 0 logico o all'1 logico si ha un trasferimento di informazioni o il completamento di una azione. Di solito i flip-flop del tipo *master-slave* sono level-triggered.

Le caratteristiche dei flip-flop di tipo master-slave, come il flip-flop JK 7476, saranno discusse nel Bugbook II e quindi, per adesso non vi chiederemo di usarli. Rimandiamo ad un capitolo successivo tutta la discussione sui flip-flop di tipo JK.

LATCH 7475

Un altro flip-flop molto usato è il latch 7475, la cui configurazione dei pin è la seguente,

7475

Nel chip 7475 ci sono quattro latch, due dei quali vengono abilitati al pin 4 e gli altri al pin 13. Se collegate il pin 4 al pin 13 potete abilitarli tutti e quattro contemporaneamente.

Avrete certamente notato che, per i pin 4 e 13, abbiamo usato il termine "abilitare" e non il termine "clock"; lo abbiamo fatto per la ragione che ogni latch bistabile del 7475 si comporta come una porta AND con memoria. Quando i pin 4 e 13 sono allo stato logico 1, il dato digitale ad ogni ingresso D passa attraverso il latch 7475 è abilitato, noi diciamo che è un follower (inseguitore), con l'uscita Q che insegue l'ingresso D.Il delay di propagazione del latch 7475 è di circa 30 ns, che è tre volte più grande del delay di propagazione di una porta TTL normale come la 7400.

La differenza tra il latch 7475 e la porta AND stà nel fatto che, anche quando l'ingresso di abilitazione passa dall'1 allo 0 logico, il valore dell'ingresso D resta bloccato, e quindi il latch 7475 ha una memoria che la porta AND 7408 non ha.

Osservate che il chip 7475 non ha né l'ingresso PRESET né l'ingresso CLEAR. La tabella della verità di questo chip, fornita dalla Texas Instruments Incorporated, è la seguente:

logic

FUNCTION TABLE
(Each Latch)

INPUTS		OUTPUTS	
D	G	Q	$\bar{Q}$
L	H	L	H
H	H	H	L
X	L	Q_0	$\bar{Q}_0$

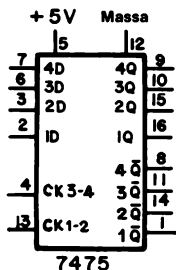
H = high level, L = low level, X = irrelevant
Q_0 = the level of Q before the high-to-low transition of G

description

These latches are ideally suited for use as temporary storage for binary information between processing units and input/output or indicator units. Information present at a data (D) input is transferred to the Q output when the enable (G) is high and the Q output will follow the data input as long as the enable remains high. When the enable goes low, the information (that was present at the data input at the time the transition occurred) is retained at the Q output until the enable is permitted to go high.

Possiamo riscrivere questa tabella della verità in termini di stati logici 0 e 1:

Ingressi		Uscite	
D	G	Q	$\bar{Q}$
0	1	0	1
1	1	1	0
X	0	Q_0	$\bar{Q}_0$

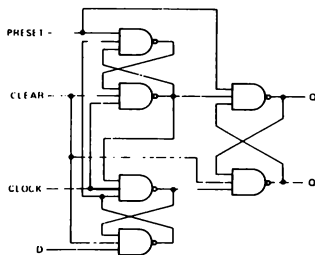
Si può capire meglio la configurazione dei pin del chip 7475 con l'aiuto del seguente schema a blocchi:

Lo schema mostra chiaramente l'esistenza di quattro latch bistabili, ciascuno dei quali ha un'uscita Q e $\overline{Q}$. Il latch 1 e 2 vengono sottoposti a clock contemporaneamente, come i latch 3 e 4.

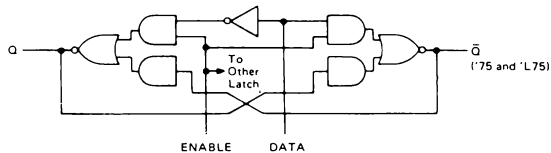
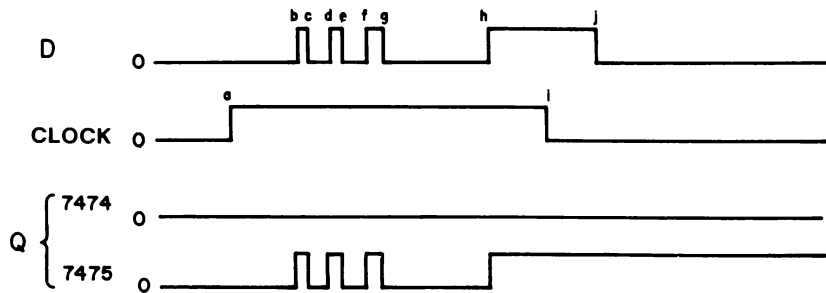
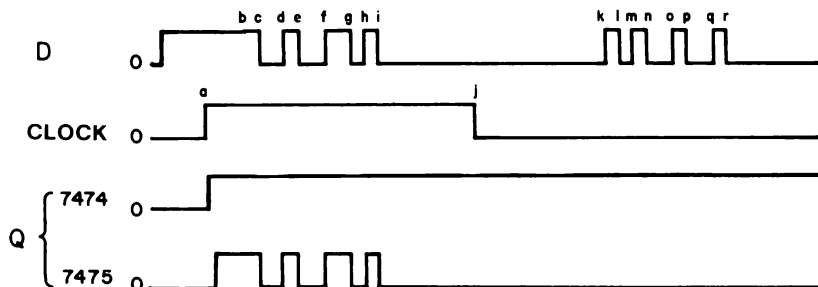
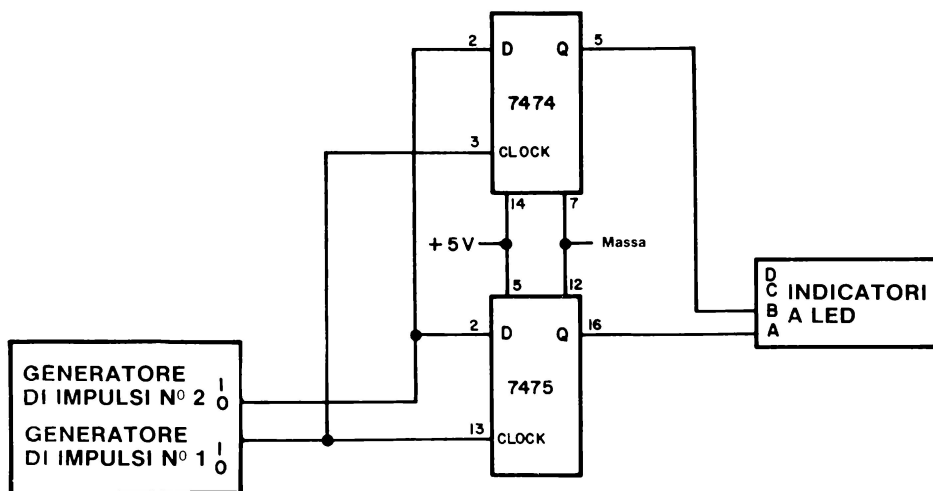
CONFRONTO TRA I LATCH 7474 e 7475

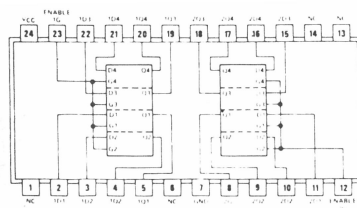
Il funzionamento dei latch 7474 e 7475 può essere riassunto nel modo seguente:

Latch 7474: *Se l'ingresso D è allo stato logico 1, l'uscita Q va o rimane allo stato logico 1 quando arriva il fronte positivo dell'impulso di CLOCK. Se l'ingresso D è allo stato logico 0, l'uscita Q va e rimane allo stato logico 0 quando arriva il fronte positivo dell'impulso di CLOCK.*

Latch 7475: *Se l'ingresso D è allo stato logico 1 quando arriva il fronte negativo dell'impulso di CLOCK, allora l'uscita rimane bloccata allo stato logico 1. Se l'ingresso D è allo stato logico 0 quando arriva il fronte negativo dell'impulso di CLOCK, allora l'uscita rimane bloccata allo stato logico 0. Quando l'ingresso di abilitazione è allo stato logico 1, l'uscita Q insegue lo stato logico dell'ingresso D con un ritardo di circa 30 ns.*

I due latches usano due combinazioni di porte completamente diverse. Il latch 7474 usa sei porte NAND a 3 ingressi,

'74–DUAL D WITH CLEAR AND PRESET

mentre il latch 7475 usa due porte AND-OR-INVERTITORE e un invertitore,

Si può vedere la vera differenza tra i due latch considerando i diagrammi temporali che si hanno quando vengono sottoposti a clock contemporaneamente. Per esempio, nei diagrammi temporali riportati qui sotto, l'uscita del latch 7474 rimane allo stato 0 logico perchè il suo ingresso D era allo 0 logico quando è sopraggiunto il fronte positivo dell'impulso di clock (punto a). Invece il 7475 "insegue" l'ingresso D fino a ché il clock rimane all'1 logico. Il bloccaggio dell'ingresso D nel 7475 avviene quando arriva il fronte negativo dell'impulso di clock (punto i).

Dal seguente diagramma temporale risulta chiaro come l'uscita del 7475 insegua l'ingresso

solo quando l'ingresso di clock è all'1 logico. Il circuito che userete per eseguire questi tipi di esperimenti è il seguente:

LATCH 74100

Il latch 74100 è un chip di 24 pin che contiene otto latch bistabili i quali si comportano come i latch 7475. La configuarzione dei pin di questo chip è:

74100

Notate che non ci sono uscite complementate Q e che vengono abilitati quattro latch per volta dai pin 12 e 23. Questi latch sono "inseguitori" come i latch 7475. La tabella della verità e la configurazione logica sono le stesse del 7475:

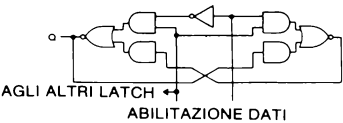
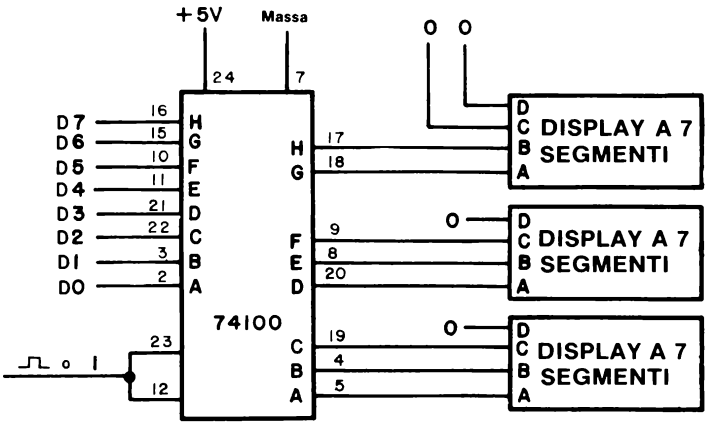
logic

FUNCTION TABLE
(Each Latch)

INPUTS		OUTPUTS	
D	G	Q	$\bar{Q}$
L	H	L	H
H	H	H	L
X	L	Q_0	$\bar{Q}_0$

H = high level, X = irrelevant
Q_0 = the level of Q before the high-to-low transition of G

description

These latches are ideally suited for use as temporary storage for binary information between processing units and input/output or indicator units. Information present at a data (D) input is transferred to the Q output when the enable (G) is high and the Q output will follow the data input as long as the enable remains high. When the enable goes low, the information (that was setup at the data input at the time the transition occurred) is retained at the Q output until the enable is permitted to go high

I microcomputer oggi più diffusi sono quelli a 8 bit e il chip 74100 va molto bene per queste macchine. Per esempio, l'uscita del data bus bidirezionale dell'MMD-1 può essere bloccata e visualizzata su tre display a sette segmenti che sono collegati a formare un display ottale a tre digit,

Quindi, se pensate di usare questo chip con un microcomputer, vi consigliamo di prendere in considerazione il fatto che ogni ingresso ha bisogno di un fan-in di 1, cioè 1,6 mA. Assicuratevi allora che ci sia un buffer/driver tra il microcomputer e il 74100 in modo di poter gestire la corrente che va da ogni ingresso del 74100. In un capitolo successivo discuteremo più dettagliatamente di queste cose.

LATCH/DISPLAY HEWLETT-PACKARD

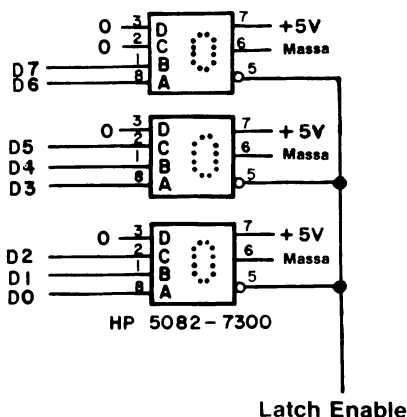
Un latch molto usato ed interessante è il latch contenuto nella serie di indicatori numerici 5082-7300 della Hewelett-Packard. Abbiamo già parlato brevemente di questi indicatori, e in questo momento ci interessano quelle caratteristiche del latch che riguardano il display numerico.

Le caratteristiche del latch a quattro bit sono essenzialmente identiche a quelle del latch bistabile dei chip 7475 e 74100. L'unica differenza è che uno 0 logico abilita la serie di indicatori numerici 5082-7300. La tabella della verità per ogni bit del latch di display numerico è:

Ingressi		Uscita
Bit d'ingresso	Latch enable	Bit sul Display
0	0	0
1	0	1
X	1	Q_0

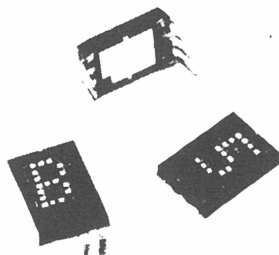
dove Q_0 è lo stato logico del bit visualizzato prima del passaggio dal valore basso a quello alto dell'ingresso di abilitazione del latch.

Detto in altre parole, i latch della serie di indicatori numerici 5082-7300 sono degli inseguitori; infatti, uno di essi quando è abilitato, insegue lo stato logico del dato in ingresso; quando è disabilitato, blocca lo stato logico che esisteva quando è giunto il fronte positivo del segnale di abilitazione.

Nel seguito riportiamo pari pari una parte della documentazione riguardante il display numerico, fornita dal costruttore. Uno degli usi per cui ci sentiamo di raccomandare questi display è come bus di monitor ottale a tre digit per gli otto bit del data bus bidirezionale del microcomputer MMD-1

NUMERIC and HEXADECIMAL INDICATORS

5082-7300 SERIES

FEATURES

- Numeric 5082-7300/-7302
  - 0-9, Test State, Minus Sign, Blank States
  - Decimal Point
    - 7300 Right Hand D.P.
    - 7302 Left Hand D.P.
- Hexadecimal 5082-7340
  - 0-9, A-F, Base 16 Operation
  - Blanking Control, Conserves Power
  - No Decimal Point
- DTL – TTL Compatible
- Includes Decoder/Driver with Memory
  - 8421 Positive Logic Input
- 4 X 7 Dot Matrix Array
  - Shaped Character, Excellent Readibility
- Standard .600 inch X .400 inch Dual-in-Line Package including Contrast Filter
- Categorized for Luminous Intensity
  - Assures Uniformity of Light Output from Unit to Unit within a Single Category

DESCRIPTION

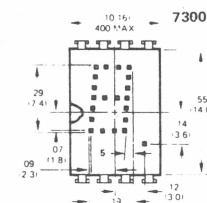
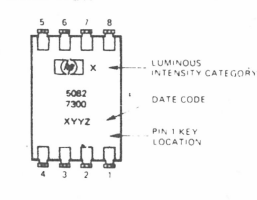
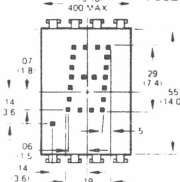
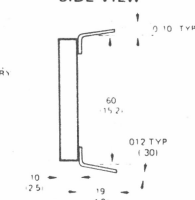
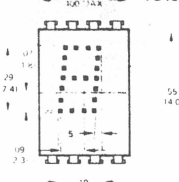
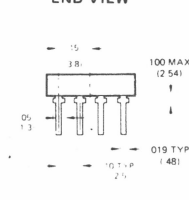
The HP 5082-7300 series solid state numeric and hexadecimal indicators with on-board decoder/driver and memory provide a reliable, low-cost method for displaying digital information.

The 5082-7300 numeric indicator decodes positive 8421 BCD logic inputs into characters 0-9, a "-" sign, a test pattern, and four blanks in the invalid BCD states, The unit employs a right-hand decimal point. Typical applications include point-of-sale terminals, instrumentation, and computer systems.

The 5082-7302 is the same as the 5082-7300, except that the decimal point is located on the left-hand side of the digit.

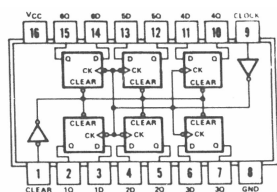
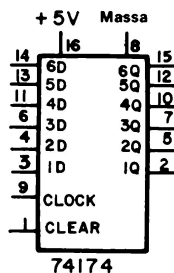
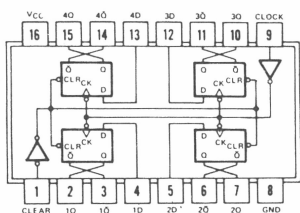
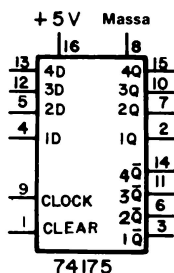
The 5082-7340 hexadecimal indicator decodes positive 8421 logic inputs into 16 states, 0-9 and A-F. In place of the decimal point an input is provided for blanking the display (all LED's off), without losing the contents of the memory. Applications include terminals and computer systems using the base-16 character set.

The 5082-7304 is a "±1" overrange character, including decimal point, used in instrumentation applications.

PACKAGE DIMENSIONS

PIN	FUNCTION 5082-7300 and 7302 Numeric	5082-7340 Hexadecimal
1	Input 2	Input 2
2	Input 4	Input 4
3	Input 8	Input 8
4	Decimal point	Blanking control
5	Latch enable	Latch enable
6	Ground	Ground
7	V_{cc}	V_{cc}
8	Input 1	Input 1

NOTES:
1. Dimensions in inches and (millimeters).
2. Unless otherwise specified, the tolerance on all dimensions is ±.015 inches.

FLIP-FLOP DI TIPO D, POSITIVE-EDGE-TRIGGERED 74174 e 74175

I flip-flop di tipo D dei chip 74174 e 74175 hanno le stesse caratteristiche del flip-flop di tipo D 7474, di cui abbiamo già parlato. La sola differenza è che nel 74174 ci sono sei flip-flop di tipo D e che nel 74175 ce ne sono 4,

Tutti i flip-flop dei chip 74174 e 74175 sono positive-edge-triggered, ognuno di essi ha l'ingresso CLEAR, ma non il PRESET. L'ingresso CLEAR pulisce tutti i sei o i quattro flip-flop contemporaneamente e, allo stesso modo, l'ingresso CLOCK di ogni chip, sottopone a clock gli ingressi D di tutti i sei, o quattro, flip-flop contemporaneamente. Questa caratteristica è illustrata di seguito. La tabella della verità di questo chip è la seguente:

FUNCTION TABLE
(EACH FLIP-FLOP)

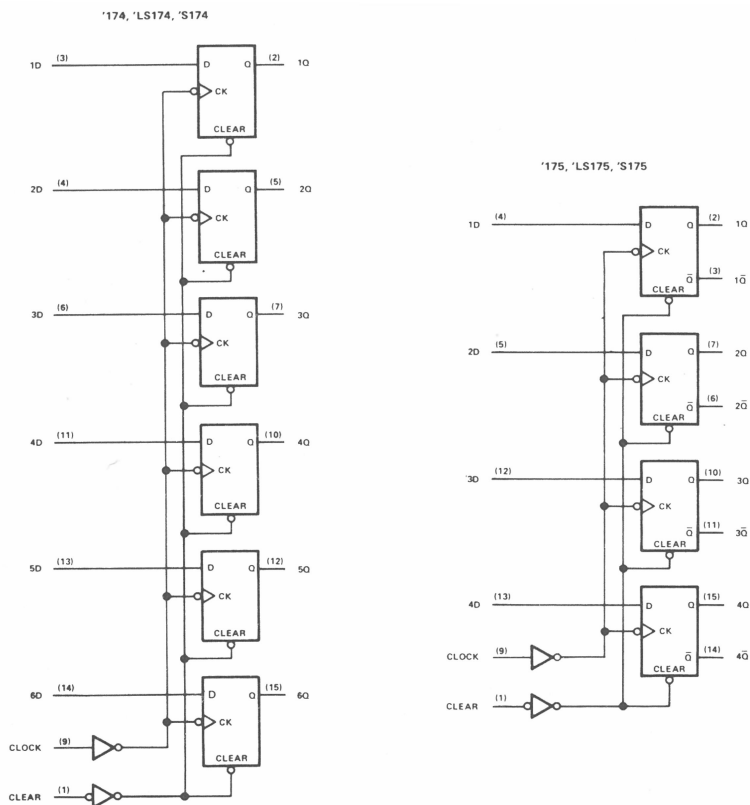
INPUTS			OUTPUTS	
CLEAR	CLOCK	D	Q	$\bar{Q}$†
L	X	X	L	H
H	↑	H	H	L
H	↑	L	L	H
H	L	X	Q_0	$\bar{Q}_0$

H = high level (steady state)
L = low level (steady state)
X = irrelevant
↑ = transition from low to high level
Q_0 = the level of Q before the indicated steady-state input conditions were established.
† = '175, 'LS175, and 'S175 only

Nei seguenti schemi a blocchi funzionali, presi dal TTL Data Book della Texas Instruments Incorporated, sono illustrati tutti i particolari degli ingressi CLEAR e CLOCK:

functional block diagrams

'174, 'LS174, 'S174

'175, 'LS175, 'S175

Il simbolo ad ogni ingresso CK di tutti i flip-flop ha il seguente significato:

Il gruppo di flip-flop che vengono contemporaneamente sottoposti a clock e puliti, funziona come un *registro.* Possono memorizzare simultaneamente tanti bit di informazione quanti sono i flip-flop del registro. Per esempio, il chip 74175 funziona come registro a 6 bit. Il concetto di registro è molto importante quando si interfaccia un microcomputer e quindi ne riparleremo in un capitolo successivo.

INTRODUZIONE AGLI ESPERIMENTI

I seguenti esperimenti vi permetteranno di verificare le differenze tra due flip-flop molto comuni, il flip-flop di tipo D 7474 positive-edge-triggered e il latch 7475. Nell'Esperimento n. 5, troverete un circuito molto usato che vi permetterà di eseguire passo passo un programma per microcomputer.

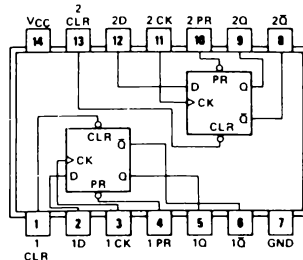
Gli esperimenti che farete possono essere così riassunti:

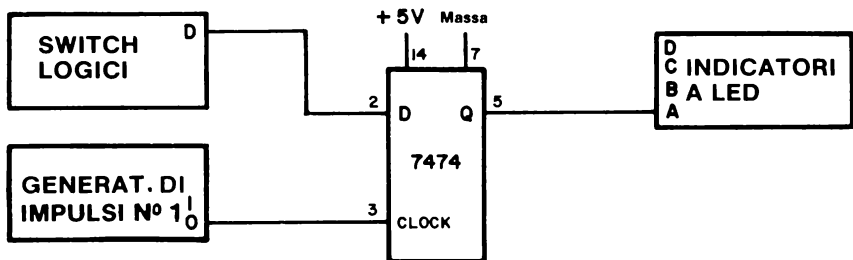
Esperimenti N.	Commento
1	Mostrate il funzionamento del flip-flop di tipo D 7474 positive-edge-triggered.
2	Mostrate come funzionano gli ingressi RESET e CLEAR del flip-flop di tipo D 7474 positive-edge-triggered.
3	Mostrate il funzionamento del latch 7475.
4	Confrontate il funzionamento del flip-flop di tipo D 7474 positive-edge-triggered col funzionamento del latch 7475.
5	Mostrate un semplice circuito single-step basato sul flip-flop 7474, che vi mette in grado di eseguire passo passo, un ciclo per volta, un programma per il microcomputer 8080A.
6	Mostrate come funziona il latch/display della Hewlett-Packard.

ESPERIMENTO N. 1

Scopo

Lo scopo di questo esperimento è di mostrare il funzionamento di un flip-flop di tipo D 7474 positive-edge-triggered quando viene usato come latch.

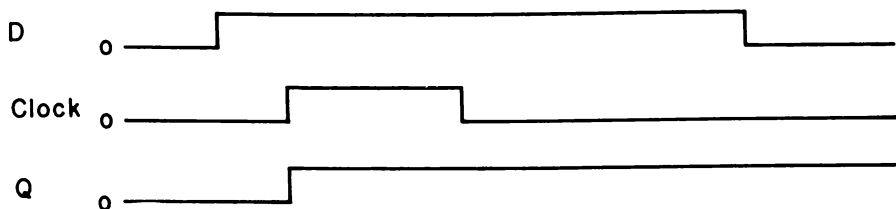
Configurazione dei pin del circuito integrato

7474

Schema del circuito

Passo 1

Collegate il circuito mostrato qui sopra. Potreste avere delle difficoltà ad usare contemporaneamente un paio di generatori di impulsi, ma potreste aiutarvi con due cacciaviti.

Passo 2

Alimentate il breadboard e mettete l'ingresso dello switch logico D all'1 logico. Adesso premete e rilasciate l'ingresso del generatore d'impulsi di clock N.1.

Dopo questa operazione, l'indicatore a LED dovrebbe accendersi. Adesso rimettete lo switch logico D allo 0 logico. Il diagramma temporale di queste operazioni è il seguente:

Dapprima l'ingresso D passa all'1 logico e ci rimane fino a che l'ingresso di clock va dallo 0 logico all'1 logico e poi torna allo 0 logico. Infine l'ingresso D torna allo 0 logico. L'uscita Q del flip-flop viene agganciata quando arriva il fronte positivo dell'impulso di clock.

Passo 3

Premete e rilasciate l'ingresso di clock del generatore di impulsi N.1. Siccome adesso l'ingresso D è allo 0 logico, dovreste vedere l'indicatore a LED spegnersi e rimanere spento. L'avete bloccato allo 0 logico.

Passo 4

Premete l'ingresso di clock del generatore di impulsi N.1 e tenetelo premuto. Fate scattare più volte l'ingresso D. Come si comporta l'indicatore a LED?

Adesso potete rilasciare il generatore di impulsi N.1. Noi abbiamo visto che l'indicatore a LED resta spento e questo perchè l'ingresso D era allo 0 logico quando è giunto il fronte positivo dell'impulso di clock. Questo è dunque lo stato logico che è rimasto bloccato e non importa cosa è successo dopo all'ingresso D.

Passo 5

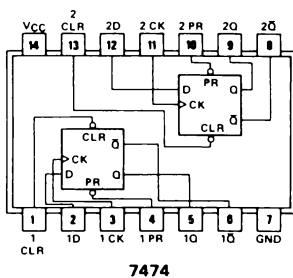
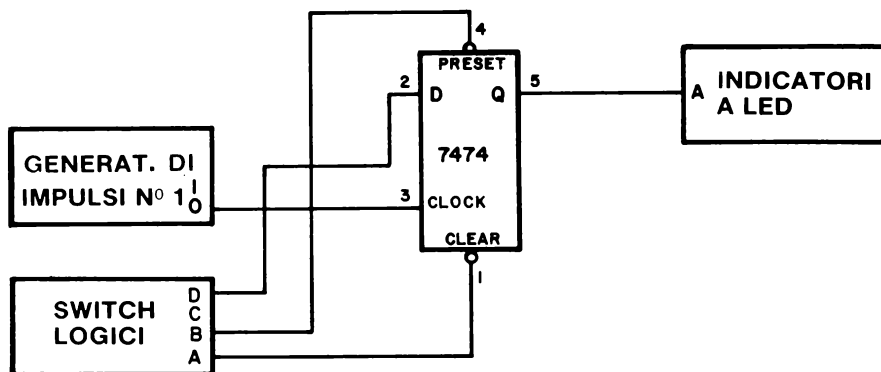
Mettete lo switch logico nella posizione 1 logico. Premete il generatore d'impulsi N.1 e osservate che l'indicatore a LED cambia e si porta allo stato logico 1. Tenendo premuto il generatore di impulsi N.1, commutate lo switch logico. *Non dovreste vedere alcun cambiamento nello stato logico degli indicatori a LED, che si trova all'1 logico, nonostante abbiate cambiato più volte il valore dell'ingresso D.*

Tutti questi esperimenti confermano che il flip-flop 7474 blocca l'uscita quando giunge il fronte positivo dell'impulso di clock.

ESPERIMENTO N. 2

Scopo

Lo scopo di questo esperimento è di mostrare il funzionamento degli ingressi asincroni CLEAR e PRESET del flip-flop di tipo D 7474.

Configurazione dei pin del circuito integrato

Schema del circuito

Passo 1

Collegate il circuito. Non dimenticate di collegare i +5V al pin 14 e la massa al pin 7 del chip 7474. Alimentate il breadboard.

Passo 2

Mettete l'ingresso CLEAR all'1 logico e l'ingresso PRESET allo 0 logico e vedrete che gli indicatori a LED si accendono; questo significa che l'uscita Q del flip-flop 7474 è allo stato logico 1.

Adesso premete e rilasciate il generatore di impulsi N.1 e commutate più volte lo switch D cercando di portare l'uscita del 7474 allo 0 logico. Ci riuscite?

La risposta è no, infatti l'ingresso asincrono PRESET prevarica l'ingresso sincrono D e l'ingresso di CLOCK.

Passo 3

Mettete l'ingresso PRESET all'1 logico, l'ingresso CLEAR allo 0 logico e vedrete che l'indicatore a LED si spegne. Adesso l'uscita del flip-flop 7474 è allo stato logico 0.

Premete e rilasciate il generatore di impulsi N. 1 e commutate più volte lo switch D cercando di ottenere dal 7474 un'uscita all'1 logico. Ci riuscite?

Anche questa volta la risposta è no, infatti l'ingresso asincrono CLEAR prevarica l'ingresso D e l'ingresso di clock.

Passo 4

Mettete sia l'ingresso PRESET che CLEAR all'1 logico. Adesso premete e rilasciate il generatore di impulsi N. 1, commutate più volte lo switch D e dimostrate che il 7474 si comporta come nell'esperimento N. 1 di questo capitolo.

Ricordate che un ingresso non collegato è allo stato logico 1. Qual'era dunque lo stato degli ingressi PRESET e CLEAR quando avete fatto le prove previste dall'Esperimento N. 1?

Erano entrambi all'1 logico perchè non erano collegati.

Passo 5

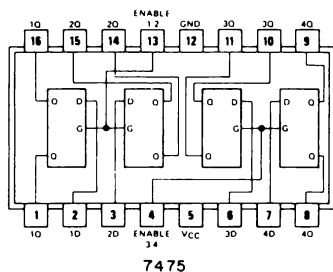
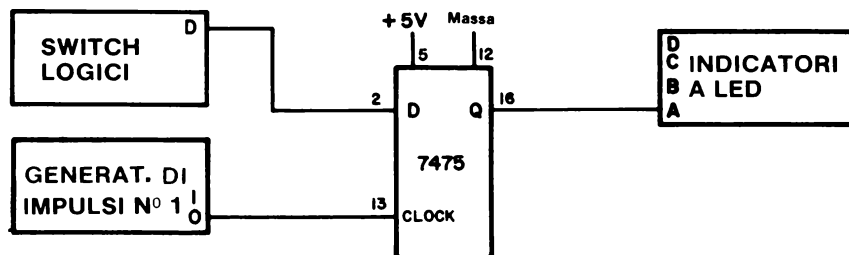
Mettete allo 0 logico sia l'ingresso PRESET che CLEAR. Cosa vedete?

Non c'è una risposta unica. Esiste sempre una condizione logica, ma non è definibile a priori.

ESPERIMENTO N. 3

Scopo

Lo scopo di questo esperimento è di mostrare il funzionamento di un semplice latch 7475.

Configurazione dei pin del circuito integrato

Schema del circuito

Passo 1

Senza togliere il flip-flop 7474, collegate il circuito che vi abbiamo illustrato sul breadboard.

Passo 2

Eseguite le seguenti azioni *nell'ordine dato:*

1. Attivate lo switch logico D (cioè passatelo dallo stato logico 0 allo stato logico 1) e osservate i cambiamenti dell'indicatore a LED. Cosa vedete?

2. Tenendo premuto il generatore d'impulsi, attivate ancora lo switch D. Che relazione c'è tra l'ingresso D e l'uscita Q?

   Per ottenere ciò, l'ingresso di CLOCK deve essere all'1 logico?

3. Tenendo lo switch logico all'1 logico, premete e rilasciate il generatore d'impulsi di CLOCK. Q cambia stato?

   Che relazione c'è tra l'ingresso D e l'uscita Q, dopo l'impulso di clock tra D e Q?

4. Mettete lo switch logico allo 0 logico, premete e rilasciate ancora il generatore di impulsi di clock. Q cambia stato?

   Che relazione c'è adesso tra D e Q?

5. Ripetete i punti 3 e 4, ma dopo aver dato un impulso di clock (premere e rilasciare) commutate lo switch logico. L'uscita Q cambia? Potete dedurre che il latch tipo 7475 "memorizza" il dato presente all'ingresso D quando l'impulso di clock torna allo 0 logico?

Si può quindi dedurre che il latch 7475 si comporta come un inseguitore (follower) quando l'ingresso di clock, o abilitatore, è all'1 logico. Quando diciamo "inseguitore" intendiamo che l'uscita Q è uguale all'ingresso D.

Discussione

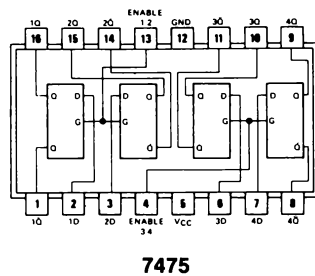
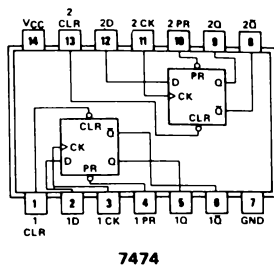
Il latch 7475 è una delle memorie da 1 bit più usate ed è possibile trovarlo inserito in molti dei più recenti circuiti integrati a larga scala di integrazioni (LSI), compreso il microprocessor 8080A, la memoria a lettura/scrittura 2102 e il chip porta di I/O a 8 bit, 8212. La Intel Corporation costruisce tutti questi chip.

Questo spiega l'importanza di capire le caratteristiche dei diversi tipi di flip-flop. Lavorando coi microprocessor abbiamo notato che i due flip-flop più importanti sono il latch e il positive-edge-triggered.

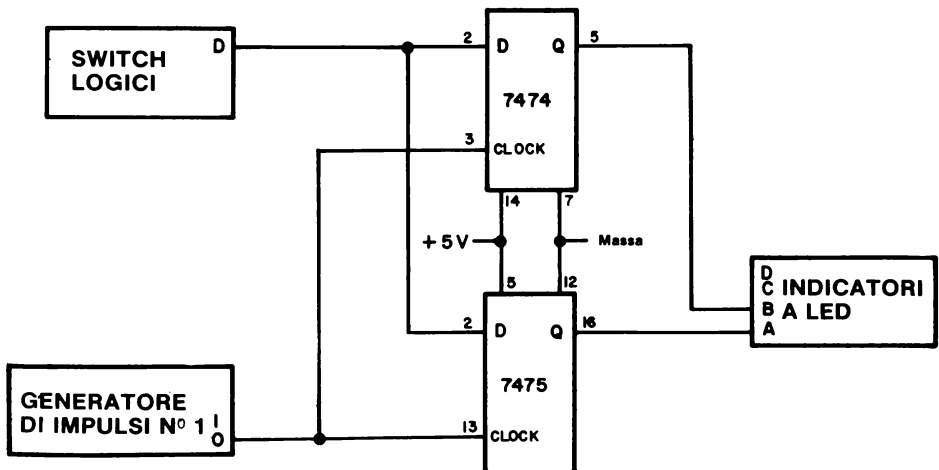
ESPERIMENTO N. 4

Scopo

Lo scopo di questo esperimento è di confrontare il funzionamento del flip-flop tipo D 7474 e del latch 7475.

Configurazioni dei pin dei circuiti integrati

7474

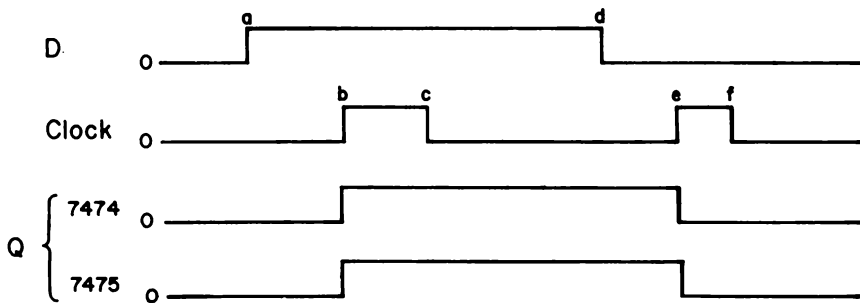
7475

Schema del circuito

Passo 1

Collegate sul vostro breadboard il circuito mostrato. Non dimenticate di alimentare entrambi i chip 7474 e 7475.

Tutti e due i chip vengono sottoposti contemporaneamente a clock dal generatore di impulsi N. 1 ed entrambi ricevono contemporaneamente l'ingresso D dallo switch logico D.

Passo 2

Alimentate il breaboard ed eseguite l'esperimento mostrato nel diagramma temporale sotto riportato. Osservate le uscite dei due latch e controllate se sono quelle indicate.

Noi abbiamo visto che le uscite del 7474 e del 7475 sono uguali a quelle indicate. Se avete avuto delle difficoltà a capire il diagramma temporale, seguite le istruzioni date nel Passo 3, altrimenti passate direttamente al Passo 4.

Passo 3

Per applicare il diagramma temporale degli ingressi D e clock, eseguite le seguenti azioni:

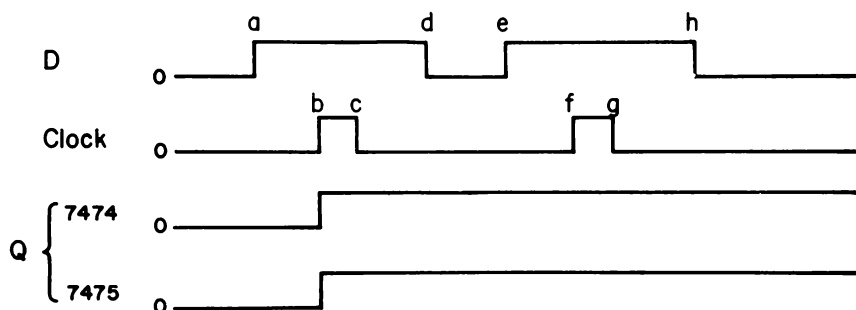
- Assicuratevi che le uscite di entrambi i latch siano allo 0 logico. Se non lo sono, mettete a 0 lo switch logico e premete e rilasciate il generatore di impulsi N. 2.
- Quando giungete al punto *a*, mettete a 1 lo switch logico e lasciatevelo fino a quando non arrivate al punto *d*.
- Al punto *b*, premete il generatore di impulsi N. 1 per un secondo e poi rilasciatelo al punto *c*.
- Aspettate qualche secondo e poi riportate l'ingresso dello switch logico D allo 0 logico.
- Aspettate ancora 1 o 2 secondi e quindi premete l'ingresso di clock del generatore di impulsi N. 1 e poi rilasciatelo (punti *e* e *f*).

- Osservate cosa succede alle uscite del 7474 e del 7475 ai punti *b* ed *e*. Al punto *b*, si accenderanno entrambi gli indicatori a LED. Al punto *e*, si spengono.

Abbiamo descritto a parole il diagramma temporale del Passo 2, ma vi invitiamo a studiarlo in modo che in seguito possiate eseguire degli esperimenti consultando direttamente i soli diagrammi temporali. Le industrie elettroniche fanno largo uso di questi diagrammi temporali. Esistono delle rappresentazioni speciali che forniscono i diagrammi temporali nello stesso istante in quindici punti diversi di un circuito.

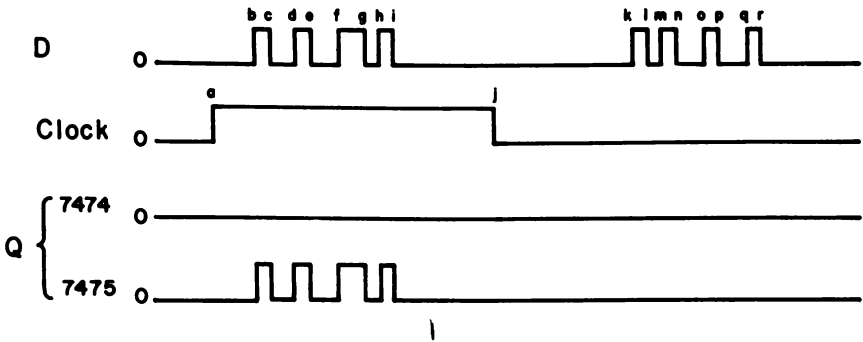
Avrete certamente notato che abbiamo chiamato "ingresso di clock del generatore d'impulsi", il generatore di impulsi N. 2. Perchè?

Passo 4

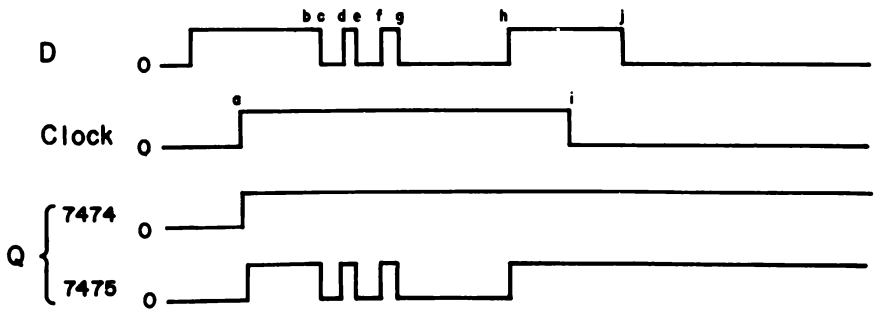
Il diagramma temporale del secondo esperimento è il seguente:

Eseguite le azioni illustrate per l'ingresso D e per gli ingressi di clock, e dimostrate che al punto *b* si accendono entrambi gli indicatori a LED e che rimangono accesi anche in seguito. Fino ad ora i due latch si sono comportati sempre nello stesso modo perchè non è stato ancora fatto nessun esperimento che ne mettesse in luce le differenze. Pulite entrambi i flip-flop, cioè portateli allo 0 logico.

Passo 5

I diagrammi temporali del terzo esperimento sono riportati di seguito. Dovreste vedere che lo *stato logico del 7474 non cambia durante tutto l'esperimento,* ma rimane allo 0 logico. Al contrario l'uscita del 7475 "segue" l'ingresso D quando l'ingresso di clock è all'1 logico.

Perchè il 7474 resta allo 0 logico?

L'uscita del 7474 rimane allo 0 logico perchè l'ingresso D è allo 0 logico quando giunge il fronte positivo dell'impulso di clock e quindi il 7474 resta bloccato allo 0 logico in cui è al punto *a*.

Passo 6

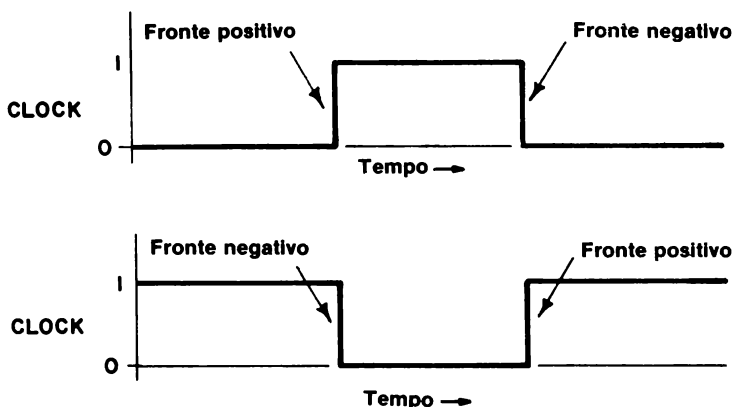
I diagrammi temporali del quarto ed ultimo esperimento sono:

Adesso potete vedere che sia il flip-flop 7474 che il flip-flop 7475 sono bloccati allo stato logico 1. Perchè?

Daremo la risposta nel Passo 7.

Passo 7

La differenza tra i due latch può essere sintetizzata come segue:

- *Il flip-flop 7474 memorizza dati presenti all'ingresso D in corrispondenza del fronte positivo del clock*
- *Il latch 7475 "insegue" i dati presenti all'ingresso D ogniqualvolta l'ingresso di clock è abilitato. Memorizza l'ingresso D in corrispondenza del fronte negativo del clock.*

Nel caso vi dimentichiate il significato di *fronte positivo e fronte negativo,* valgono le figure seguenti:

Un impulso di clock positivo, transazione di 0 logico ad 1 logico ed ancora 0 logico, è rappresentata nella prima figura; nella seconda è indicato un impulso di clock negativo.

Per il flip-flop 7474, lo stato logico dell'ingresso D in corrispondenza del fronte positivo dell'ingresso di clock, è lo stato che è poi memorizzato. In più, l'uscita del 7475 insegue le variazioni dell'ingresso D ogniqualvolta l'ingresso di clock è allo stato logico 1.

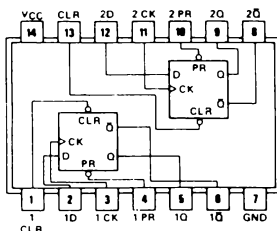
Possiamo ora spiegare tutti i differenti diagrammi temporali con facilità. Iniziamo dal diagramma del passo 5. Al punto *a*, il fronte positivo, del clock, il 7474 memorizza uno 0 e rimane allo stato 0. Al punto *b* fino al punto *i*, l'uscita del 7475 segue l'ingresso D finchè l'ingresso di clock è ad 1. Al punto *j*, il 7475 memorizza uno stato logico 0 e resta 0. Alla fine, ai punti *k* fino ad *r*, non succede nulla finchè l'ingresso di clock non è allo stato logico 1. Il 7475 non segue queste escursioni logiche.

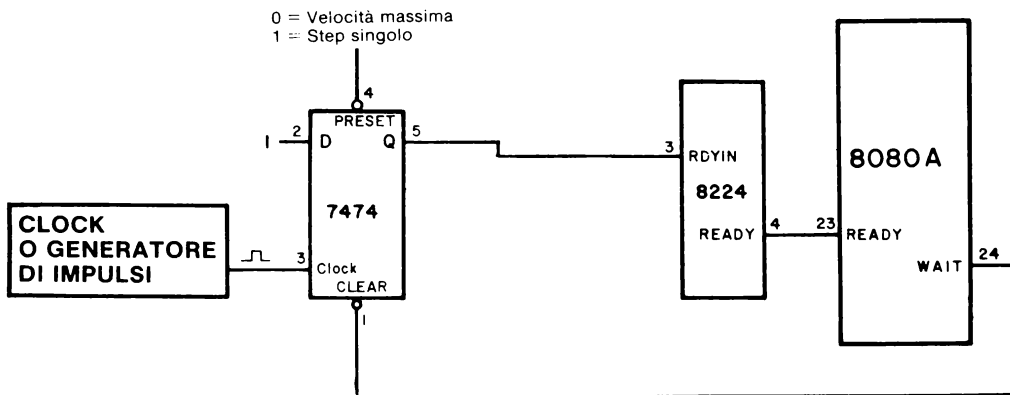
Nel Passo 6, si è in una situazione similare. Il 7474 memorizza un 1 logico al presentarsi di tale stato logico all'ingresso D durante il fronte positivo del clock. Anche il 7475 è bloccato ad 1 logico al presentarsi dell'1 all'ingresso D durante il fronte negativo del clock.

ESPERIMENTO N. 5

Scopo

Lo scopo di questo esperimento è mutare e controllare un circuito single-step per il microcomputer MMD-1.

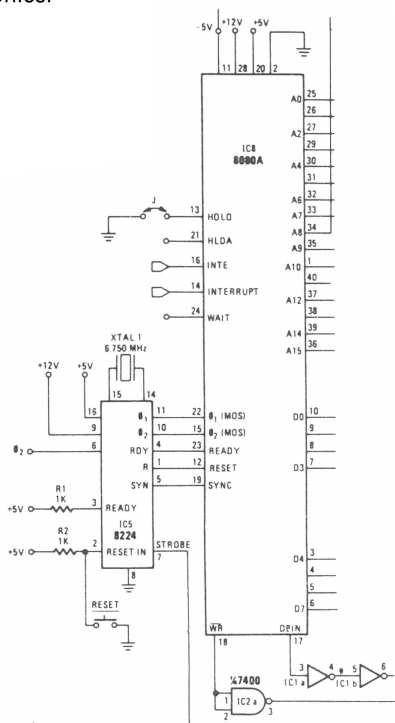
Configurazione dei pin del circuito integrato

7474

Schema del circuito

Il circuito precedente è presentato per concessione del Sig. William Dalton, studente del Virginia Polytechnic Institute e State University.

Programma

Indirizzo di memoria LO	Byte di istruzione	Codice mnemonico	Descrizione
000	074	INR A	Incrementa di 1 il contenuto dell'accumulatore
001	323	OUT	Poni in uscita il contenuto dell'accumulatore
002	002	002	Codice della porta 2
003	303	JMP	Salto alla locazione di memoria data dai due bytes seguenti
004	000	—	Byte LO
005	003	—	Byte HI

Passo 1

Tolta l'alimentazione dal microcomputer, montate il circuito fornito nello schema. Il collegamento tra il chip 8224 ed il chip 8080A già esiste sulla piastra. Di seguito è indicato lo schema dei collegamenti come sono sulla piastra dell'MMD-1 per concessione della rivista Radio-Electronics.

Il pin 24 del chip 8080A, l'uscita WAIT, è fornita come pin senza saldatura del breadboard BP-25. Il pin 3 del chip 8224, meglio conosciuto come RDYIN, è accessibile nella parte superiore del circuito stampato in corrispondenza del resistore pull-up da 1 kΩ. Potete scegliere se fare dei collegamenti clip-lead o se saldare i fili al di sotto della piastra del circuito stampato.

Una volta che avete individuato gli ingressi RDYIN e WAIT, il resto diventa facile. Collegate il circuito illustrato nello schema e non dimenticate i pin di alimentazione, Vcc e GND (massa), del chip 7474.

Passo 2

Adesso collegate l'alimentazione. Il pin 4 del chip 7474 dovrebbe essere allo 0 logico, che corrisponde al modo di operare alla velocità di 750 kHz. Caricate il programma che abbiamo illustrato e poi premete il tasto G. Dovrebbero accendersi tutti gli indicatori a LED della porta 2.

Passo 3

Mettete il pin 4 del chip 7474 all'1 logico. In questo modo è possibile ottenere il funzionamento passo passo del microcomputer MMD-1. Premendo e rilasciando il generatore di impulsi, o collegando a un circuito di clock che lavora alla frequenza di circa 1 kHz, dovreste vedere gli indicatori a LED della porta 2 che si incrementano lentamente.

Passo 4

Se avete un Outboard LR-27 latch/display ottale, collegatelo col data bus bidirezionale e collegate l'ingresso STB allo 0 logico. L'Outboard vi permetterà di vedere quello che succede nel data bus ad ogni singolo passo del programma.

Eseguite il programma passo-passo. Notate che potete riprodurre tutto il programma, ma che, *tra il byte 002 e il byte 303 c'è un byte in più;* è il byte del terzo *ciclo di macchina* della istruzione OUT. Durante questo ciclo di macchina, che è generato dal microcomputer, il dato contenuto nell'accumulatore viene messo in uscita sul data bus ed è generato un impulso di sincronizzazione di OUT. Discuteremo queste cose con più dettaglio a partire dal capitolo 17.

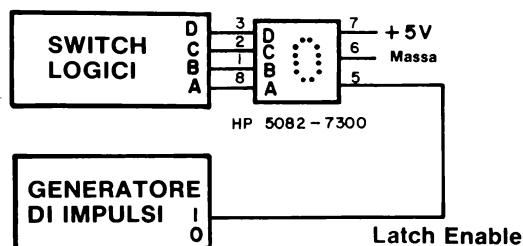
Passo 5

Se volete modificare il programma o premere un qualsiasi tasto della tastiera del microcomputer MMD-1, dovete fare in modo che il pin 4 del chip 7474 sia allo 0 logico. Per usare i tasti della tastiera bisogna che il microcomputer funzioni ad alta velocità. Nella Keyboard Executive EPROM esiste un loop di delay che serve per fornire una logica antirimbalzo ai tasti meccanici. Alla velocità di clock di 750 kHz, il delay è di almeno 10 ms., quindi, se il microcomputer è rallentato da un clock esterno e funziona a 1 Hz per *ciclo macchina,* il delay aumenta e diventa di circa 40 o 50 minuti. Riparleremo di questo argomento in un successivo capitolo.

ESPERIMENTO N. 6

Scopo

Lo scopo di questo esperimento è di mostrare il funzionamento del latch/display della Hewlett-Packard.

Schema del circuito

Passo 1

Collegate questo circuito e assicuratevi che i collegamenti di alimentazione siano giusti perchè, se per caso li invertiste, distruggereste il display.

Passo 2

Mettete gli switch logici nei seguenti stati: A=0, B=1, C=1 e D=0. Vedete qualche cambiamento sul display?

La risposta dovrebbe essere no.

Passo 3

Adesso premete e rilasciate il generatore di impulsi. Che numero appare sul display?

Sei.

Passo 4

Adesso mettete gli switch logici nei seguenti stati: A=1, B=1, C=0 e D=0. Premete e rilasciate il generatore di impulsi. Sul display dovrebbe apparire un tre.

Passo 5

Adesso collegate l'ingresso abilitatore del latch (latch enable) che si trova al pin 5 del display all'uscita "0" del generatore di impulsi. Spostate gli switch logici passando tutti i valori tra DCBA=0000 e DCBA=1001. Cosa vedete sul latch/display?

Il display "segue" tutti i cambiamenti degli switch ed abbiamo visto apparire in sequenza tutti i decimali, dallo 0 al 9.

Passo 6

Quando il latch è abilitato il display "segue" i dati in ingresso e quando il latch è disabilitato, viene bloccato il dato presente al pin 5 quando giunge il fronte negativo dell'ingresso di clock. Questo latch/display è positive-edge-triggered?

No. Le sue caratteristiche operative assomigliano a quelle del latch 7475?

Si, se si esclude il fatto che lo 0 logico abilita il latch/display, mentre il latch è abilitato dall'1 logico.

DOMANDE RIEPILOGATIVE

Le seguenti domande servono ad aiutarvi a ripassare le caratteristiche dei flip-flop e dei latch.

1. È possibile costruire un qualsiasi tipo di flip-flop usando un appropriato numero di porte logiche di qualsiasi tipo?

2. Che differenza c'è tra un ingresso asincrono ed un ingresso sincrono di un flip-flop?

3. È possibile dotare di logica antirimbalzo uno switch meccanico usando un paio di porte NAND a 2 ingressi o un paio di porte NOR a 2 ingressi. Quali sono migliori e perchè? Date una risposta breve.

4. Cos'è una race condition?

5. Quali sono le differenze basilari tra i flip-flop 7474 e 7475?

6. Cosa significa il termine delay di propagazione?

RISPOSTE

1. Si. Vi raccomandiamo però, ogni volta che sia possibile, di usare le porte NAND.

2. Gli ingressi asincroni possono influenzare lo stato dell'uscita di un flip-flop indipendentemente dall'azione dell'ingresso di clock, invece gli ingressi sincroni influenzano lo stato dell'uscita solo sotto comando di un segnale di clock.

3. È preferibile l'uso di due porte NAND. Le porte NOR hanno bisogno di molta più corrente e inoltre è molto difficile abbassare un livello logico al potenziale di massa, come richiedono le porte NOR, che alzarlo al livello potenziale di +5V, come succede nel caso delle porte NAND.

4. Si ha un race condition quando due segnali che dipendono dal tempo "corrono" provocando una reazione od un gruppo di reazioni. Il segnale che lavora meno rapidamente può causare degli effetti indesiderati, come degli impulsi detti "glitch", che mancano assolutamente nei circuiti che funzionano correttamente.

5. Il flip-flop 7474 acquisisce l'informazione presente all'ingresso D quando giunge il fronte positivo dell'impulso di clock. In tutti gli altri punti dell'impulso di clock, non succede nulla. Il latch 7475 "insegue" lo stato logico dell'ingresso D fino a che l'ingresso di clock si mantiene all'1 logico. Quando giunge il fronte negativo dell'impulso di clock, l'informazione presente all'ingresso D, viene agganciata e bloccata. Quando l'ingresso di clock passa all'1 logico, non succede nulla.

6. Il delay di propagazione è il tempo richiesto perchè un segnale logico attraversi un singolo dispositivo logico o un gruppo di dispositivi logici. I segnali non passano istantaneamente attraverso gli elementi logici.

CAPITOLO 12

DECODIFICATORI

INTRODUZIONE

Questo capitolo vi introdurrà all'utilizzo dei decodificatori, come il 7442, 74154 e 74155. Sia il 7442 che il 74LS155 si trovano sulla piastra del circuito stampato del microcomputer MMD-1, e vengono usati per decodificare i canali di I/O dei blocchi di memoria. Nel capitolo 17 verrà discusso l'uso dei decodificatori nelle applicazioni relative all'interfacciamento.

OBIETTIVI

Alla fine di questo capitolo sarete in grado di:

- Convertire un numero binario di 8 bit in un numero esadecimale di due digit.
- Dare la definizione di codice esadecimale.
- Elencare gli equivalenti di alcuni dei più importanti caratteri alfanumerici codificati in ASCII.
- Convertire la parola di 8 bit di ogni carattere del codice ASCII in codice esadecimale.
- Dire quali sono le differenze dei decodificatori 7442 e 74154.
- Dire cosa significa codificare e decodificare.
- Scrivere la tabella della verità di un tipico decodificatore.

CODICI DIGITALI

Come abbiamo già detto nel capitolo 1, un *codice digitale* può essere definito come un sistema di simboli che rappresentano il valore dei dati e che formano un linguaggio che un computer o un circuito digitale può interpretare e usare. I codici digitali possono essere considerati come dei linguaggi digitali che permettono di memorizzare, trattare e comunicare delle informazioni. Così come esistono numerose lingue, esistono numerosi codici digitali, che però possono venir suddivisi in quattro importanti categorie:

Categoria I — Codici impiegati per descrivere la circuiteria elettronica e le varie operazioni digitali. Esempio: codice binario.

Categoria II — Codici usati per convertire i numeri decimali da 0 a 9 in forma digitale. Esempi: codice binario, binario decimale codificato (BCD), codice Gray.

Categoria III — Codici usati per convertire i numeri decimali, le lettere dell'alfabeto, i simboli e le operazioni in forma digitale. Esempi: codice ASCII, codice EBCDIC, codice Baudot.

Categoria IV — Codici istruzioni usati dai computer, dai minicomputer e dai microcomputer, che permettono a queste macchine di eseguire una prefissata sequenza di operazioni. Esempi: codice istruzione per IBM 370, codice istruzione per PDP 8/E, codice istruzione per 8080A, codice istruzione per Motorola 6800, codice istruzione per Fairchild F8.

Rispetto all'argomento trattato in questo Bugbook, vi dovrebbero interessare maggiormente il codice binario, il codice ottale, il codice binario decimale codificato, il codice ASCII e il codice istruzioni per il microprocessor 8080A.

Nel capitolo 1 abbiamo parlato del codice binario e del codice ottale in questo capitolo vi insegneremo il codice esadecimale, il binario decimale codificato (BCD), il codice ASCII ed inoltre i decodificatori 7442 e 74154.

CODICE ESADECIMALE

Il codice esadecimale fa riferimento al *sistema di calcolo esadecimale*, un sistema che si fonda su una *base* o *radice* di 16. Il sistema di calcolo esadecimale fa uso di sedici simboli differenti: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E e F. Come avete già visto per i numeri ottali è possibile convertire anche i numeri esadecimali in numeri binari.

Numeri esadecimali	Numeri binari
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000

Numeri esadecimali	Numeri binari
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111
10	0001 0000
11	0001 0001
12	0001 0010
13	0001 0011
14	0001 0100
15	0001 0101
16	0001 0110
17	0001 0111
18	0001 1000
19	0001 1001
1A	0001 1010
1B	0001 1011
1C	0001 1100
1D	0001 1101
1E	0001 1110
1F	0001 1111
20	0010 0000
30	0011 0000
40	0100 0000
50	0101 0000
60	0110 0000
70	0111 0000
80	1000 0000
90	1001 0000
A0	1010 0000
B0	1011 0000
C0	1100 0000
D0	1101 0000
E0	1110 0000
F0	1111 0000
FF	1111 1111

Abbiamo diviso i numeri binari di 8 bit in due gruppi di quattro per aiutarvi a capire come si esegue la conversione di un numero esadecimale in numero binario, ma di solito non viene riportato lo spazio tra i due gruppi di quattro bit.

Per convertire un numero binario di 8 bit in codice esadecimale seguite queste istruzioni nel seguente ordine:

1. Scrivere il numero binario completo di 8 bit.
2. Dividere il numero binario di 4 bit.
3. Sostituire il corrispondente digit esadecimale - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E e F - ad ogni gruppo di 4 bit, dopo di che avrete convertito un numero binario di 8 bit in un numero esadecimale di due digit.

Per esempio, prendiamo il numero binario di 8 bit

10011101

Primo, dividerlo in due gruppi di 4 bit,

1001 1101

Infine, sostituire il corrispondente digit esadecimale ad ogni gruppo di quattro bit,

9 D

La risposta esatta è $9D_H$, dove l'indice "$_H$" rappresenta il sistema di calcolo esadecimale (in inglese Hexadecimal). Spesso troverete i numeri esadecimali espressi nella forma 9DH perchè si incontrano delle difficoltà a stampare l'indice con una teletype o un terminale.

BINARIO DECIMALE CODIFICATO

Uno dei codici più usati per codificare i numeri decimali è il codice 8 4 2 1, detto anche *binario decimale codificato (BCD).* Questa tabella mette in confronto i numeri decimalí con il sistema binario decimale codificato:

Numero decimale	Binario decimale codificato
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

È chiaro che il binario decimale codificato è un codice molto semplice di quattro bit. che inizia da 0000 e termina a 1001; ogni decimale è codificato con quattro bit, come mostrato di seguito per il numero decimale 9:

	2^3	2^2	2^1	2^0
Binario decimale codificato	= 1	0	0	1
Notazione per chip IC	= D	C	B	A
Altre notazioni	= MSB			LSB

Osservate che il decimale 9 è uguale a $1 \times 2^3 + 1 \times 2^0$ dove abbiamo usato i pesi di ogni bit del numero BCD.

Sui circuiti integrati troverete spesso l'annotazione DCBA e sarà così anche nel caso dei contatori binari e decimali. La lettera "D" rappresenta il bit più significativo (MSB = most significant bit) dei quattro bit del numero BCD, mentre la lettera "A" rappresenta il bit meno significativo (LSB = least significant bit).

Il binario decimale codificato è un codice, per rappresentare i numeri decimali, molto facile e conveniente, ma è ridondante. Bisogna infatti usare troppi bit binari per rappresentare un numero decimale; per esempio, il numero decimale 4095_{10}, ha bisogno di ben sedici bit in BCD mentre ne bastano dodici per rappresentarlo in binario.

$4095_{10} = 111111111111_2 = 0100\ \ 0000\ \ 1001\ \ 0101$ in BCD

4 0 9 5

CODICE PER DISPLAY A SETTE SEGMENTI

Un sistema molto usato per visualizzare il numero del binario decimale codificato (BCD) è quello impiegante un display che contiene sette segmenti disposti in modo da poter rappresentare i numeri decimale da 0 a 9 quando la combinazione BCD viene applicata ad un circuito integrato *decoder/driver* (decodificatore/pilota) accoppiato al display. Vorremmo adesso sottolineare che il display a sette segmenti può essere usato per visualizzare i numeri esadecimali ammesso che vi ricordiate l'equivalente esadecimale dei sei simboli che appaiono sul display.

Sul TTL Data Book della Texas Instruments sono riportati le seguenti informazioni:

IDENTIFICAZIONE DEL SEGMENTO

DESIGNAZIONI NUMERICHE E VISUALIZZAZIONI RISULTANTI

Osservate quello che viene visualizzato sul display per i numeri da 10 a 15. Basandoci su queste osservazioni possiamo comporre la tabella che converte il codice esadecimale nei simboli che appaiono sul display a sette segmenti:

Numeri esadecimali	Numero sul display a sette segmenti
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	⊏
B	⊐
C	⊔
D	≡
E	⊢
F	blank

Se voleste usare il codice esadecimale, dovreste sostituire il decoder/driver 7447 con il Fairchild 9374 o un Signetics 8T74, infatti questi chip producono i caratteri A, B, C, D, e E al posto dei simboli che abbiamo visto prima. Però dovete mettere a massa il pin 3 di questi chip perchè, come il latch/display della Hewlett-Packard, hanno un latch al loro interno.

CODIFICARE E DECODIFICARE (ENCODE AND DECODE)

Sebbene la maggior parte dei circuiti digitali operi con due stati logici e quindi esegua le operazioni usando l'aritmetica binaria, alcuni individui, per ragioni storiche o di altro tipo, preferiscono lavorare con l'aritmetica decimale, cioè rappresentando tutti i numeri con gli interi 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9 e le potenze in base 10. Considerando che al mondo esiste anche gente di questo tipo, diventa vantaggioso *codificare* un numero decimale in forma binaria, mantenere tutta la circuiteria che lavora su numeri binari che ne risultano e, infine, quando c'è bisogno di fornire l'uscita, *decodificare* i numeri binari in forma decimale. Siccome un computer di solito codifica i numeri decimali e poi decodifica i numeri binari, uno strumento di laboratorio ha bisogno di decodificare solo le funzioni binarie.

I termini *codificare* e *decodificare* possono essere definiti nel modo seguente:

Codificare	Usare un codice, di solito composto di numeri binari, per rappresentare un singolo carattere o un gruppo di caratteri di un messaggio. Passare da un codice digitale ad un altro. Se i codici sono molto diversi, il processo è detto conversione di codici[2].
Decodificare	Usare un codice per riconvertire una precedente codifica. Individuare il significato di un set di impulsi che descrive un'istruzione, un comando, una operazione da compiere[2].

CODICI ALFANUMERICI

Certe volte può essere necessario non limitare il processo di codifica e decodifica ai soli numeri decimali, ma potrebbe essere importante poter codificare tutto l'alfabeto inglese in forma digitale, come si fa per un terminale stampante (teletype-writer) o video (CRT display). Quando poi si è codificato l'alfabeto inglese, restano ancora delle posizioni del *codice alfanumerico* libere e allora, di solito, si codificano anche altri simboli quali, !, @, #, $, %, ¢, &, *, (,), -, +, =, ", ', etc., ai quali si aggiungono alcune operazioni di tastiera come "ritorno colonna 1", "back space", "avanzamento rapido (shift)", "ritorno carrello", "spazio", etc. *Comunque l'oggetto di una codifica digitale è il codificare in forma digitale i numeri decimali, l'alfabeto, le funzioni di controllo, le operazioni di tastiera, e/o i simboli,* cosa certamente fattibile negli Stati Uniti, in Europa o in America Latina, ma molto difficoltosa quando vengono usati, nelle comunicazioni, dei simboli non standard (Cina, Mondo Arabo, etc.).

CODICE ASCII

Il codice alfanumerico più usato nella trasmissione delle ventisei lettere dell'alfabeto, dei dieci numeri decimali, di ventotto simboli, e per diverse funzioni di controllo tastiera, è conosciuto come *codice ASCII*, cioè *American Standard Code for Information Interchange*. Il codice ASCII è un codice binario a sette bit che può potenzialmente codificare $2^7=128$ diverse unità di informazioni.

Nel seguito vi diamo metà tabella del codice ASCII, cioè 64 delle parole di sette bit del codice ASCII, quelle per le lettere dell'alfabeto Romano, per i numeri e per i simboli.

Le altre 64 parole del codice ASCII codificano le lettere minuscole e le funzioni di tastiera.

	Codice ASCII			Codice ASCII	
Carattere	Binario	Ottale*	Carattere	Binario	Ottale*
@	1000000	300	space	0100000	240
A	1000001	301	!	0100001	241
B	1000010	302	"	0100010	242
C	1000011	303	#	0100011	243
D	1000100	304	$	0100100	244
E	1000101	305	%	0100101	245
F	1000110	306	&	0100110	246
G	1000111	307	'	0100111	247
H	1001000	310	(	0101000	250
I	1001001	311	)	0101001	251
J	1001010	312	*	0101010	252
K	1001011	313	+	0101011	253
L	1001100	314	,	0101100	254
M	1001101	315	-	0101101	255
N	1001110	316	.	0101110	256
O	1001111	317	/	0101111	257
P	1010000	320	0	0110000	260
Q	1010001	321	1	0110001	261
R	1010010	322	2	0110010	262
S	1010011	323	3	0110011	263
T	1010100	324	4	0110100	264
U	1010101	325	5	0110101	265
V	1010110	326	6	0110110	266
W	1010111	327	7	0110111	267
X	1011000	330	8	0111000	270
Y	1011001	331	9	0111001	271
Z	1011010	332	:	0111010	272
[	1011011	333	;	0111011	273
\	1011100	334	<	0111100	274
]	1011101	335	=	0111101	275
^	1011110	336	>	0111110	276
←	1011111	337	?	0111111	277

* Il codice ottale corrispondente al codice ASCII nel quale l'ottavo bit, il più significativo, è stato posto all'1 logico. Questo è il codice ASCII che vedrete con più frequenza. Per convertire nel codice ASCII a 7 bit, basta costituire con uno 0 il bit più significativo.

Altre due parole del codice ASCII che si incontrano frequentemente sono due operazioni di tastiera, il SALTO RIGA (LINE FEED) e il RITORNO CARRELLO (CARRIAGE RETURN):

LF 0001010 012
CR 0001101 015

Il tasto RETURN fa ritornare il carrello senza terminare la riga sulla stampante, mentre il tasto LINE FEED fa avanzare la stampante o il carrello di una riga.

Notate che il codice ASCII è ancora più inefficiente del binario decimale codificato. Per esempio, il numero decimale 9 corrisponde a 011101 in codice ASCII (a 7 bit) mentre in BCD equivale a 1001. In effetti però la massima efficienza non l'aspetto più importante di un codice digitale. Altre proprietà, come la compatibilità, l'utilità e la facilità di uso sono, in certe situazioni, molto più importanti.

Per esempio, il codice ASCII è importante perchè è usato ovunque, infatti è compatibile con telescriventi, display e molti altri tipi di stampanti alfanumeriche e terminali video.

Nella tabella precedente, il bit meno significativo (LSB) è il bit più a destra della parola ASCII di 7 bit, mentre il più significativo (MSB) è quello più a sinistra. Altri codici alfanumerici molto diffusi sono il *codice Selectric* e l'*Extended Binary Coded Decimal Interchange Code (codice EBCDIC)* e vengono usati entrambi dalle macchine IBM. La Fairchild Semiconductor costruisce dei chip MOS che convertono l'ASCII in Selectric o in EBCDIC e viceversa [Chip Fairchild numero 3512A e 3514A/3514B].

CONVERSIONE DI CODICE

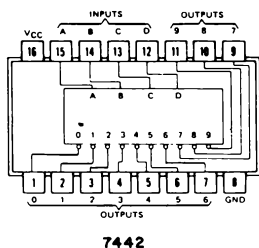
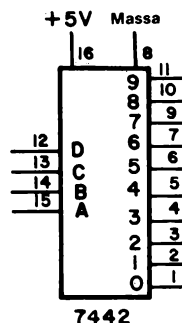
Alle Nazioni Unite gli interpreti traducono da una lingua all'altra e così le rappresentanze di più di 125 Paesi possono comunicare tra loro. Anche in elettronica digitale esiste l'esigenza di passare da un codice all'altro; questo processo è detto *conversione di codice*, cioè cambiamento di un gruppo di bit che rappresentano un carattere in un certo codice al corrispondente gruppo di bit che rappresentano, in un altro codice, lo stesso carattere. Si può schematicamente rappresentare una conversione di codice nel seguente modo:

ed ha il seguente tipo di tabella della verità:

Ingressi						Uscite				
F	E	D	C	B	A	E'	D'	C'	B'	A'

dove le righe e le colonne contengono i relativi 0 e 1 logici. In questa tabella della verità ci sono 64 ingressi, ma però non è sempre necessario usarli tutti.

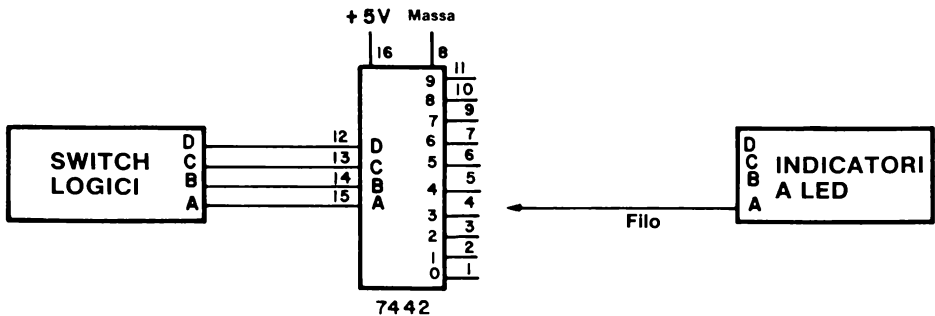
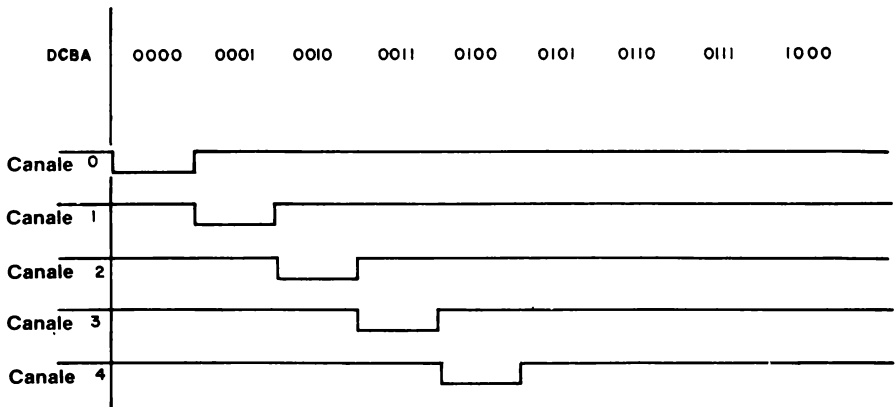
DECODIFICATORE 7442

Il circuito integrato 7442 è un *decodificatore da 4 a 10 linee* che converte la parola BCD di quattro bit in un 0 logico su una sola uscita delle dieci possibili. Il circuito integrato ha solo sedici pin, la maggior parte dei quali fungono da uscita.

7442

7442

La tabella della verità di questo chip è la seguente:

Ingressi	Uscite
DCBA	0 1 2 3 4 5 6 7 8 9
0 0 0 0	0 1 1 1 1 1 1 1 1 1
0 0 0 1	1 0 1 1 1 1 1 1 1 1
0 0 1 0	1 1 0 1 1 1 1 1 1 1
0 0 1 1	1 1 1 0 1 1 1 1 1 1
0 1 0 0	1 1 1 1 0 1 1 1 1 1
0 1 0 1	1 1 1 1 1 0 1 1 1 1
0 1 1 0	1 1 1 1 1 1 0 1 1 1
0 1 1 1	1 1 1 1 1 1 1 0 1 1
1 0 0 0	1 1 1 1 1 1 1 1 0 1
1 0 0 1	1 1 1 1 1 1 1 1 1 0
1 0 1 0	1 1 1 1 1 1 1 1 1 1
1 0 1 1	1 1 1 1 1 1 1 1 1 1
1 1 0 0	1 1 1 1 1 1 1 1 1 1
1 1 0 1	1 1 1 1 1 1 1 1 1 1
1 1 1 0	1 1 1 1 1 1 1 1 1 1
1 1 1 1	1 1 1 1 1 1 1 1 1 1

A questo punto risulterà certamente chiaro perchè questo decodificatore è detto *decodificatore 1 su 10,* infatti per ognuno dei dieci numeri, da 0 a 9, solo una delle uscite è allo 0 logico, mentre le altre sono tutte all'1 logico. I decodificatori 7441, 7442, 7445, 74141 e 74145, sono tutti di questa categoria. La principale differenza tra questi decodificatori nel tollerare diverse tensioni di uscita che sono, nei diversi casi, +5V, +15V, +30V o +70V. Questi decodificatori vengono anche detti *decodificatori da BCD a decimale* o *decodificatori/pilota da BCD a decimale* e, in particolare, il termine pilota (driver) si usa solo per quelli che tollerano tensioni o correnti più alte del normale.

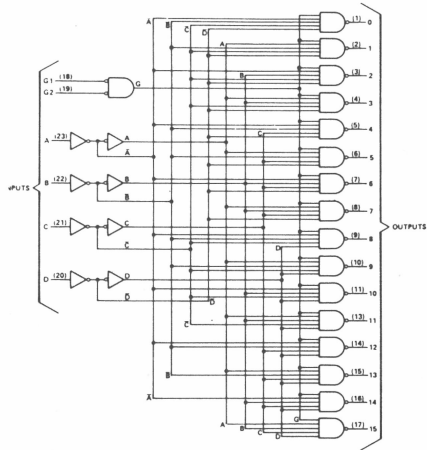
Di seguito è illustrato un circuito molto semplice che impiega un decodificatore 7442:

Si può capire il comportamento di questo circuito guardando un suo diagramma temporale:

Quando DCBA=0001, il canale 1 è allo stato 0 logico e gli altri canali sono all'1 logico. Quando DCBA=0100, il canale 4 è allo 0 logico e gli altri all'1 logico. Inoltre, se DCBA è tra 1010 e 1111, tutti i canali di uscita sono all'1 logico perchè il 7442 è solo un decodificatore da 4 a 10 linee e non da 4 a 16. Più avanti eseguirete l'esperimento di cui adesso abbiamo solo parlato.

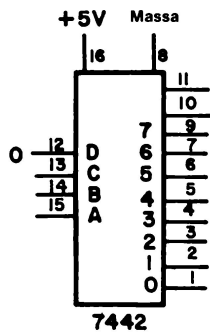
DECODIFICATORE 74154

Un decodificatore molto usato nell'interfacciamento del microcomputer è il 74154, *decodificatore da 4 a 16 linee*, che converte una parola binaria di quattro bit in uno stato logico 0 ad una sola uscita delle sedici possibili. Il circuito integrato è più grande del corrispondente 7442 a 16 pin e la tabella della verità è simile a quella del 7442 con due ingressi di abilitazione in più:

Ingressi						Uscite															
G1	G2	D	C	B	A	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	1	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
0	0	1	0	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
0	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	X	X	X	X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	X	X	X	X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	X	X	X	X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

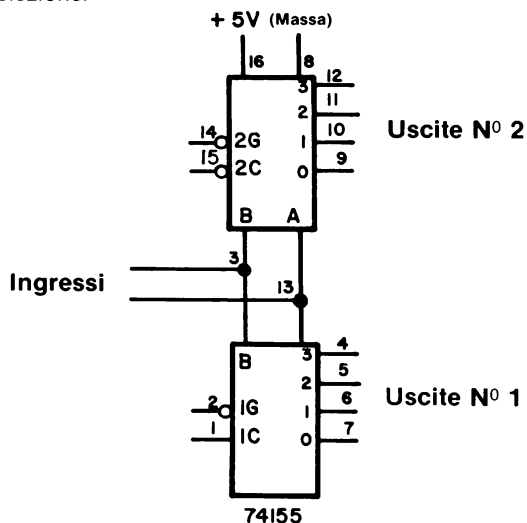
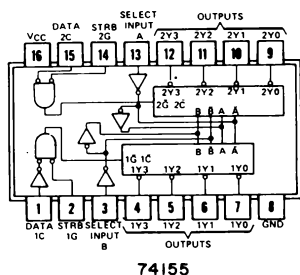
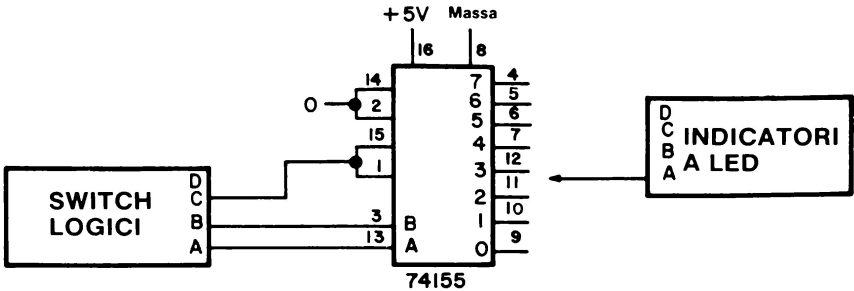
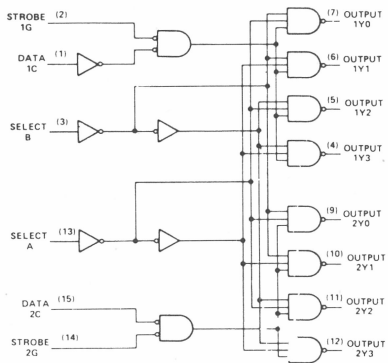
Evidentemente, sia G1 che G2 devono essere allo stato 0 logico per abilitare il chip 74154. Una "X" indica gli stati logici non rilevanti. Lo schema a blocchi del chip 7442 è illustrato nel capitolo 7. Il corrispondente schema a blocchi per il chip 74154 è il seguente:

functional block diagram and schematics of inputs and outputs

DECODIFICATORI DA 3 A 8 LINEE

Fino a qui abbiamo visto i decodificatori da 4 a 10 linee e da 4 a 16 linee, ma, mettendo l'ingresso D allo 0 logico in un chip 7442, si può creare un decodificatore da 3 a 8 linee perchè in pratica si limitano gli ingressi tra 0000 e 0111.

Ingressi CBA	Uscite 0 1 2 3 4 5 6 7
0 0 0	0 1 1 1 1 1 1 1
0 0 1	1 0 1 1 1 1 1 1
0 1 0	1 1 0 1 1 1 1 1
0 1 1	1 1 1 0 1 1 1 1
1 0 0	1 1 1 1 0 1 1 1
1 0 1	1 1 1 1 1 0 1 1
1 1 0	1 1 1 1 1 1 0 1
1 1 1	1 1 1 1 1 1 1 0

Si può trasformare il decodificatore 74154 in un decodificatore da 3 a 8 linee mettendo allo 0 logico l'ingresso D del pin 20.

DECODIFICATORE 74155

Il 74155, *doppio decodificatore da 2 a 4 linee* contiene due decodificatori da 2 a 4 linee che hanno un set comune di impulsi di selezione.

Come collegare questo chip per farlo diventare un decodificatore da 3 a 8 linee sarà indicato successivamente. Vi daremo anche la tabella della verità e il diagramma funzionale del chip; entrambe per concessione della Texas Instruments Incorporated dal suo TTL Data Book. Il chip 74155 è molto interessante e versatile. Sulla piastra del circuito stampato del microcomputer MMD-1 c'è un 74LS155 che decodifica blocchi di memoria, come la memoria a lettura/scrittura, memoria EPROM e tutte le locazioni di memoria oltre il primo K.

functional block diagram and logic

FUNCTION TABLES
2-LINE-TO-4-LINE DECODER
OR 1-LINE-TO-4-LINE DEMULTIPLEXER

INPUTS				OUTPUTS			
SELECT		STROBE	DATA				
B	A	1G	1C	1Y0	1Y1	1Y2	1Y3
X	X	H	X	H	H	H	H
L	L	L	H	L	H	H	H
L	H	L	H	H	L	H	H
H	L	L	H	H	H	L	H
H	H	L	H	H	H	H	L
X	X	X	L	H	H	H	H

INPUTS				OUTPUTS			
SELECT		STROBE	DATA				
B	A	2G	2C	2Y0	2Y1	2Y2	2Y3
X	X	H	X	H	H	H	H
L	L	L	L	L	H	H	H
L	H	L	L	H	L	H	H
H	L	L	L	H	H	L	H
H	H	L	L	H	H	H	L
X	X	X	H	H	H	H	H

FUNCTION TABLE
3-LINE-TO-8-LINE DECODER
OR 1-LINE-TO-8-LINE DEMULTIPLEXER

INPUTS				OUTPUTS							
SELECT			STROBE OR DATA	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
C†	B	A	G‡	2Y0	2Y1	2Y2	2Y3	1Y0	1Y1	1Y2	1Y3
X	X	X	H	H	H	H	H	H	H	H	H
L	L	L	L	L	H	H	H	H	H	H	H
L	L	H	L	H	L	H	H	H	H	H	H
L	H	L	L	H	H	L	H	H	H	H	H
L	H	H	L	H	H	H	L	H	H	H	H
H	L	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	L	H	H
H	H	L	L	H	H	H	H	H	H	L	H
H	H	H	L	H	H	H	H	H	H	H	L

†C = inputs 1C and 2C connected together
‡G = inputs 1G and 2G connected together
H = high level, L = low level, X = irrelevant

ALTRI DECODIFICATORI E DECODIFICATORI/PILOTA

In questo capitolo abbiamo parlato dei decodificatori che userete subito negli esperimenti di interfacciamento, ma nella serie di circuiti integrati 7400 ne esistono molti altri ed alcuni di essi sono importanti, per cui ne parleremo più tardi; invece adesso diamo subito dei brevi ragguagli sui rimanenti:

- 7446 e 7447, decodificatore/pilota da 4 a 7 linee conosciuti anche come decodificatori/pilota da BCD a sette segmenti.

  Vengono usati per accendere i singoli segmenti dei display a sette segmenti ad anodo comune come l'Opcoa SLA-1. Hanno la possibilità di far apparire i numeri 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9 ed altri simboli.

- 7448, decodificatore/pilota da 4 a 7 linee, conosciuto anche come decodificatore/pilota da BCD a sette segmenti.

  È simile ai chip 7446 e 7447 e lavora con display a sette segmenti con catodo comune come il Litronix 704.

- 7445, decodificatore/pilota da 4 a 10 linee.

  Ognuna delle sue dieci uscite può assorbire delle lampadine fino a 60 mA; la sua logica è simile a quella del 7442.

- 74141, decodificatore/pilota da 4 a 10 linee.

  Ogni uscita sopporta delle tensioni molto alte, e quindi è possibile usare i tubi Nixie®.

- 74159, decodificatore da 4 a 16 linee con uscita a open collector.

  È simile al 74154, ma ha le uscite ad open collector che ne permette l'utilizzo in sistemi a bus. In un prossimo capitolo parleremo delle tecniche di uso degli open collector nei sistemi a bus.

- 74184 e 74185A, convertitori da binario a BCD e da BCD a binario.

  Questi chip possono fare la conversione da BCD a binario e da binario a BCD in pochi nanosecondi. Per i numeri a più digit BCD si usano più chip in cascata. Anche il microcomputer può eseguire queste operazioni e se siete capaci di fare questo tipo di conversione in 40 ns, non avete bisogno di usare questi chip.

- 74138 e 74139, decodificatori:

  Il 74138 è un decodificatore da 3 a 8 linee, il 74139 è un decodificatore da 2 a 4 linee ed entrambi hanno tre ingressi di abilitazione (enable) che ne semplificano l'uso in cascata e/o la ricezione dei dati. Sono circuiti Schottky-clamped TTL MSI e possono essere usati nei sistemi ad alte prestazioni che hanno bisogno di delay di propagazione molto brevi, cosa che non è necessaria con i microcomputer a 8 bit esistenti.

TEMPI TIPICI DI ABILITAZIONE E SELEZIONE DEI DECODIFICATORI

Ogni operazione logica viene eseguita in un tempo finito. In rapporto al vostro microcomputer, la serie dei chip 7400 è molto veloce, comunque i decodificatori hanno due tempi molto importanti:

Tempo di selezione — È il tempo necessario a selezionare una specifica uscita del decodificatore dopo che gli appropriati stati logici sono stati applicati agli ingressi selezionati.

Tempo di abilitazione — È il tempo necessario ad abilitare il chip decodificatore dopo che gli appropriati stati logici sono stati applicati all'ingresso di abilitazione (enable).

I tempi di selezione ed abilitazione dei decodificatori di cui abbiamo parlato sono i seguenti:

Tipo di decodificatore	Numero del chip	Tempo di selezione, ns	Tempo di abilitazione, ns	Totale potenza dissipata, mW
Da 4 a 16 linee	74154	23	19	170
	74L154	46	38	85
	74159	24	19	170
Da 4 a 10 linee	7442	17	--	140
	74L42	34	--	70
Da 3 a 8 linee	74138	8	7	225
	74LS138	22	21	31
Da 2 a 4 linee	74S139	7,5	6	300
	74LS39	22	19	34
	74LS155	18	15	30
	74155	21	16	250
	74156	23	18	250

Per un microcomputer basato sull'8080A che opera alla frequenza 750 kHz, avete a disposizione almeno 1,33 ns per decodificare l'indirizzo del bus e abilitare un chip decodificatore come il 74154, quindi i tempi di tutti i decodificatori che abbiamo menzionato sono molto più brevi di quanto vi serva e allora si possono usare i chip L o LS per ridurre l'assorbimento di potenza di corrente dai chip LSI come l'8080A. 5 μs (cioè 5000 ns) in confronto a 30 ns, sono una eternità.

Dovreste essere capaci di localizzare, sulla piastra del circuito stampato del microcomputer MMD-1, i decodificatori 74LS155 e 74L42.

INTRODUZIONE AGLI ESPERIMENTI

I seguenti esperimenti mostrano il funzionamento di alcuni tipi di decodificatori.

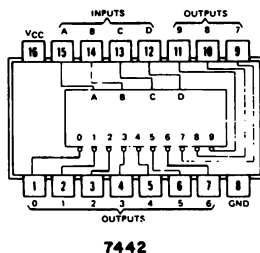
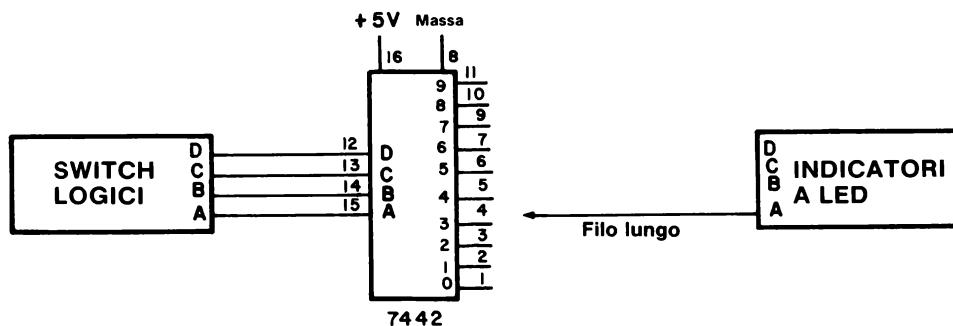
Esperimento N.	Commento
1	Mostrare il funzionamento del chip 7442 come semplice decodificatore da BCD a decimale
2	Mostrare il funzionamento del chip 7442 come decodificatore da 3 a 8 linee.
3	Mostrare il funzionamento del chip 74154 come decodificatore da 4 a 16 linee.
4	Mostrare come si usa il chip 74155 come decodificatore da 2 a 4 linee.

ESPERIMENTO N. 1

Scopo

Lo scopo di questo esperimento e di mostrare il funzionamento del chip 7442 come semplice decodificatore da BCD a decimale.

Configurazione dei pin del circuito integrato

7442

Schema del circuito

Passo 1

Studiate molto bene questo schema e osservate che il numero dei canali si trova all'interno del blocco del chip, mentre il numero del pin si trova all'esterno. Vi necessita un solo indicatore a LED che deve essere collegato con un filo abbastanza lungo per poter controllare lo stato di tutti e dieci i canali.

Passo 2

Collegate un circuito (vi bastano sette collegamenti) e alimentate il breadboard.

Passo 3

Mettete i quattro switch logici DCBA a 0 (0000) e collegate il filo lungo dell'indicatore a LED A al pin 1 del chip 7442. L'indicatore non dovrebbe accendersi.

Adesso provate gli altri canali del chip 7442. Cosa vedete?

Dovreste vedere che tutti gli altri canali di uscita sono allo stato logico 1

Passo 4

Mettete i quattro switch logici DCBA nella configurazione 0001. Adesso il canale 1 dovrebbe essere allo 0 logico. Lo è?

Nel nostro caso lo è, e tutti gli altri canali sono all'1 logico.

Passo 5

Mettete i quattro switch logici DCBA nella configuarzione 0010. Quale canale di uscita è allo 0 logico?

Il canale 2, corrispondente al pin 3, è allo 0 logico; tutti gli atri canali dovrebbero essere all'1 logico, e nel nostro caso, lo sono.

Passo 6

Quale ingresso DCBA è necessario per selezionare il canale 5? Quale pin di uscita è allo 0 logico quando avete selezionato questo canale?

Gli switch logici devono essere nella configurazione DCBA=0101. Dovrebbe essere allo 0 logico il pin 6.

Passo 7

Completate la seguente tabella della verità basandovi sui vostri esperimenti:

Numero del canale	Ingresso codificato DCBA	Uscita codificata Numero del canale 0 1 2 3 4 5 6 7 8 9
0	0 0 0 0	0 1 1 1 1 1 1 1 1 1
1	0 0 0 1	1 0 1 1 1 1 1 1 1 1
2	0 0 1 0	
3	0 0 1 1	
4	0 1 0 0	
5	0 1 0 1	
6	0 1 1 0	
7	0 1 1 1	
8	1 0 0 0	
9	1 0 0 1	

Ricordate che dovete impostare tutte le combinazioni degli switch logici e poi provare tutti i canali di uscita per trovare qual'è l'unico allo stato logico 0.

Passo 8

Cosa succede quando DCBS=1010, 1011, 1101, 1110 o 1111?

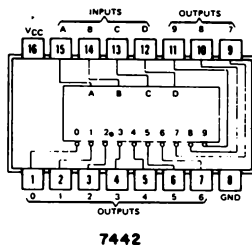
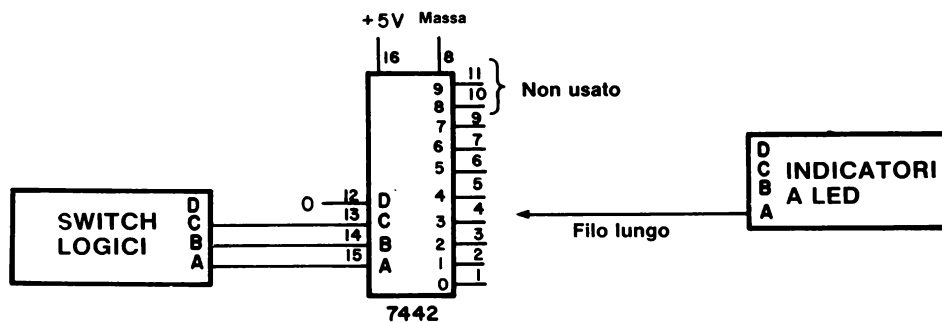
Tutti i pin di uscita sono all'1 logico perchè il 7442 è un decodificatore da BCD a decimale e non da binario a esadecimale, può quindi decodificare quattro bit in solo dieci, e non sedici, uscite uniche.

Non smontate il circuito e passate al prossimo esperimento.

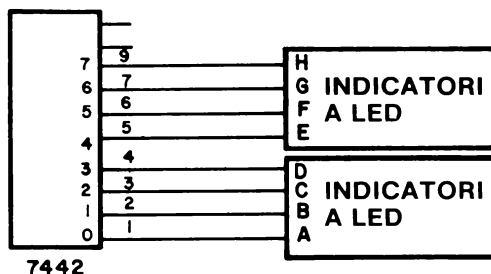
ESPERIMENTO N. 2

Scopo

Lo scopo di questo esperimento è di mostrare il funzionamento del chip 7442 come decodificatore da 3 a 8 linee.

Configurazione dei pin del circuito integrato

Schema del circuito

Passo 1

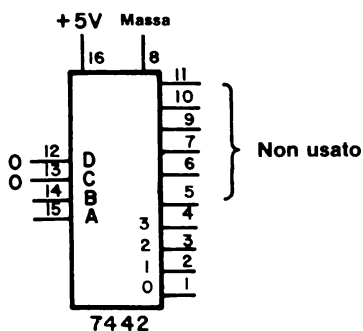
Lasciate tutto come nell'esperimento precedente, tranne l'ingresso D che metterete allo 0 logico.

Se li avete aggiungete altri indicatori a LED come specificato dal prossimo schema, così potrete controllare contemporaneamente tutti e otto i canali.

Passo 2

Variando in tutti i modi possibili gli switch logici rimasti, dimostrate che il chip 7442, con l'ingresso D allo 0 logico, funziona come decodificatore da 3 a 8 linee e che ha la seguente tabella della verità:

Numero del canale	C B A	Numero del canale 0 1 2 3 4 5 6 7
0	0 0 0	0 1 1 1 1 1 1 1
1	0 0 1	1 0 1 1 1 1 1 1
2	0 1 0	1 1 0 1 1 1 1 1
3	0 1 1	1 1 1 0 1 1 1 1
4	1 0 0	1 1 1 1 0 1 1 1
5	1 0 1	1 1 1 1 1 0 1 1
6	1 1 0	1 1 1 1 1 1 0 1
7	1 1 1	1 1 1 1 1 1 1 0

Passo 3

Che tipo di decodificatore è quello di questo schema?

12-24

Il decodificatore dello schema è un 7442 trasformato in decodificatore da 2 a 4 linee la cui tabella della verità è la seguente:

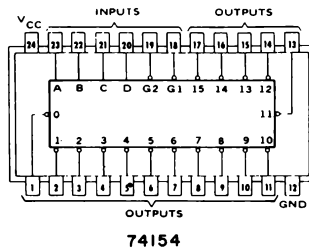
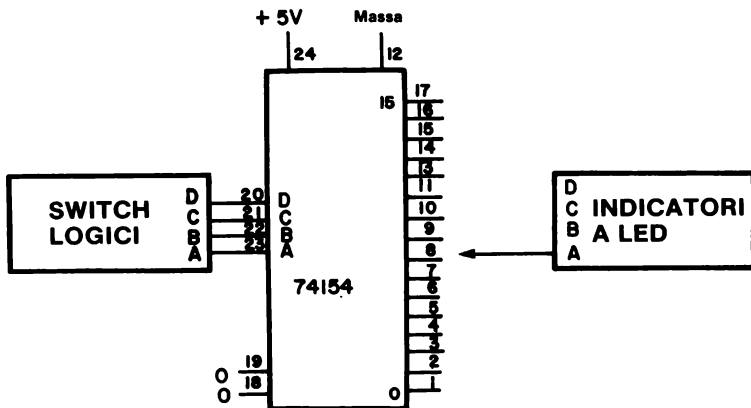
Numero del canale	B A	Numero del canale 0 1 2 3
0	0 0	0 1 1 1
1	0 1	1 0 1 1
2	1 0	1 1 0 1
3	1 1	1 1 1 0

ESPERIMENTO N. 3

Scopo

Lo scopo di questo esperimento è di mostrare il funzionamento del chip 74154 come decodificatore da 4 a 16 linee.

Configurazione dei pin del circuito integrato

74154

Schema del circuito

Passo 1

Per eseguire questo esperimento è sufficiente un indicatore a LED che sarà usato per controllare lo stato logico di ognuno dei sedici canali di uscita del decodificatore.

Collegate il circuito e alimentate il breadboard.

Passo 2

Mettete gli switch logici DCBA nella configurazione 0000 e collegate il filo dell'indicatore a LED al canale 0 (pin 1). L'indicatore non dovrebbe accendersi.

Adesso controllate gli altri quindici canali del decodificatore; dovreste vedere che sono tutti allo stato logico 1.

Passo 3

Mettete gli switch logici DCBA nella configurazione 0001, cioè mettete lo switch logico A all'1 logico. Adesso il canale 1 dovrebbe essere allo 0 logico e tutti gli altri all'1 logico. È cosí?

Si.

Passo 4

Mettete i quattro switch logici DCBA nella configurazione 0010, cioè mettete lo switch logico B all'1 logico e gli altri allo 0 logico. Quale canale di uscita è allo 0 logico?

È allo 0 logico il canale 3 (pin 3).

Passo 5

Completate la seguente tabella della verità del decodificatore da 4 a 16 linee:

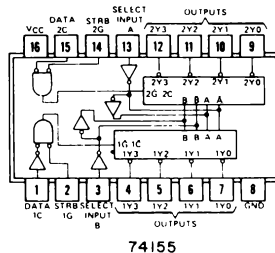
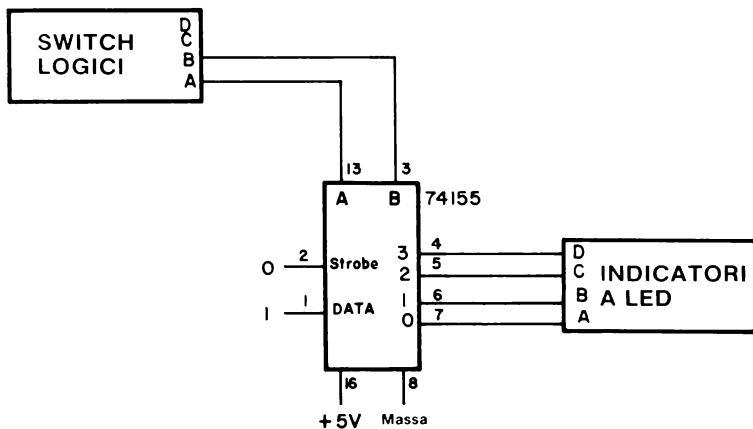
Ingressi	Uscite															
DCBA	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0 0 0 0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0 0 0 1	1	0														
0 0 1 0	1	1	0													
0 0 1 1																
0 1 0 0																
0 1 0 1																
0 1 1 0																
0 1 1 1																
1 0 0 0																
1 0 0 1																
1 0 1 0																
1 0 1 1																
1 1 0 0																
1 1 0 1																
1 1 1 0																
1 1 1 1																

ESPERIMENTO N. 4

Scopo

Lo scopo di questo esperimento è di mostrare come si usa il chip 74155 come decodificatore da 2 a 4 linee.

Configurazione dei pin del circuito integrato

74155

Schema del circuito

Passo 1

Collegate questo circuito e alimentate il breadboard.

Passo 2

Commutate gli switch logici A e B in tutti i modi possibili, controllate che i numeri appaiono sui quattro indicatori a LED e completate la seguente tabella della verità:

Ingressi	Uscite
B A	0 1 2 3
0 0	
0 1	
1 0	
1 1	

Passo 3

Qual'è la funzione dell'ingresso STROBE del chip 74155? Se non siete sicuri della risposta, mettete l'ingresso STROBE all'1 logico e controllate i canali di uscita per vedere se qualcuno di essi è allo 0 logico. Cosa vedete?

Con lo STROBE all'1 logico, tutti i canali di uscita sono all'1 logico indipendentemente dagli ingressi di selezione A e B; si dice, allora, che il chip 74155 è stato "disabilitato".

DOMANDE RIEPILOGATIVE

Le seguenti domande servono ad aiutarvi a ripassare i decodificatori e i codici digitali.

1. A quali numeri binari corrispondono i seguenti numeri esadecimali?

   a. 3C

   b. C3

   c. D3

   d. DB

   e. 00

   f. FB

   g. 76

   h. C2

2. Rispondete ancora alla domanda n. 1, ma convertendo i numeri esadecimali in codice ottale da 3 digit.

3. Convertite i seguenti caratteri dal codice ASCII a 8 bit (supponendo che il bit più significativo sia all'1 logico) in codice esadecimale.

   a. A

   b. 0 (numero decimale)

   c. 1

   d. ?

   e. LF

   f. CR

   g. spazio (blank)

4. Se l'ingresso del decodificatore 74154 fosse A=0, B=1, C=0 e D=1, quale canale di uscita sarebbe allo stato logico 0 se, sia G1 che G2 fossero allo 0 logico?

   Se G1 poi cambiasse dallo 0 logico all'1 logico, che stato logico si avrebbe sul canale di uscita? G1 e il canale di uscita sarebbero sempre allo stesso stato logico? Riuscite ad immaginare una possibilità di impiego per questa caratteristica?

RISPOSTE

1. a. 00111100
   b. 11000011
   c. 11010011
   d. 11011011
   e. 00000000
   f. 11111011
   g. 01110110
   h. 11000010

2. a. 074
   b. 303
   c. 323
   d. 333
   e. 000
   f. 373
   g. 166
   h. 302

Potete riconoscere questi byte ottali come alcune delle istruzioni più usate dell'8080A.

3. a. C1
   b. B0
   c. B1
   d. BF
   e. 8A
   f. 8D
   g. A0

4. Il canale 10. Ogni volta che G1 è allo stato 0 logico, il canale 10 è allo 0 logico. Quando G1 è all'1 logico, il canale 10 è all'1 logico. Una variazione nel tempo dei dati digitali, come un treno di impulsi di clock, che entra a G1, apparirà invariata sul canale 10. Variando gli stati logici degli ingressi di selezione DCBA, il dato variabile nel tempo può essere messo in uscita su uno qualsiasi dei sedici canali di uscita. Questo tipo di operazione è detto *demultiplexing* ed è discusso in dettaglio nel Bugbook I[0].

CAPITOLO 13

CONTATORI

INTRODUZIONE

In questo capitolo descriveremo alcuni dei più comuni tipi di contatori della famiglia di circuiti integrati 7400, che comprende anche i contatori 7490, 7492 e 7493. Imparerete a costruire un generatore di sequenze usando un contatore e un decodificatore; imparerete anche ad usare il microcomputer come un generatore di sequenze ottali.

OBIETTIVI

Alla fine di questo capitolo sarete in grado di:

- Costruire un circuito contatore a più decadi.
- Costruire e far funzionare un generatore di sequenze decimali e esadecimali.
- Dare la definizione di contatore, contatore binario e contatore decimale.
- Individuare un contatore dalle sue caratteristiche più importanti, quali il modulo, la direzione di conteggio, il sincronismo, la possibilità di avere un preset e la possibilità di collegamento in cascata.
- Descrivere il funzionamento del contatore e decade 7490 e del contatore binario 7493.
- Misurare la frequenza di un clock digitale come quello che si ha dall'Outboard LR-5.

COSA È UN CONTATORE?

Una delle operazioni più importanti dell'elettronica digitale è il *conteggio*, cioè la misurazione del numero d'impulsi di clock che si hanno in un tempo determinato. Delle quantità fisiche come resistenze, capacità, tensioni, correnti, eventi, frequenze e simili, vengono di solito trasformati in una serie di impulsi di clock il cui numero, in periodo di tempo determinato, è in qualche modo proporzionale alla grandezza delle quantità fisiche misurate. Malmstadt e Enke[9] hanno definito i termini *contatore* e *contatore binario* nel seguente modo:

Contatore — Dispositivo in grado di cambiare stato, in una determinata sequenza, quando riceve degli appropriati segnali di ingresso. L'uscita del contatore indica il numero di impulsi che sono stati applicati.(vedere anche Divisore). Un contatore è composto di flip-flop e porte e di solito l'uscita del flip-flop è sempre accessibile per poter controllare il valore del conteggio.

Contatore binario — È una interconnessione di flip-flop con un solo ingresso messo in modo tale da poter effettuare il conteggio. Ogni volta che giunge all'ingresso un impulso, il contatore cambia stato e scandisce il numero di impulsi in ingresso in modo che appaia in uscita in forma binaria. Ogni contatore può contare 2" impulsi, dove n è il numero di flip-flop.

I contatori binari possono essere usati per costruire i *contatori divisori per n*, cioè dei contatori che producono un singolo impulso di uscita ogni n impulsi di ingresso. Possono essere usati anche come *riduttori di scala* o *divisori di frequenza*, i quali riducono la frequenza di ingresso dividendola per n. Un *contatore a decade* viene definito nel seguente modo:

Contatore a decade — Contatore che contiene porte logiche e flip-flop messi in modo da permettergli di contare le decine. Dispositivo logico che ha dieci stati stabili e che può essere guidato nella scansione di questi dieci stati mediante dieci impulsi di ingresso. Di solito un contatore a decade conta, in sequenza binaria, dallo stato 0 allo stato 9 dopo di che torna allo stato 0. Alcuni li chiamano *contatori divisori-per-dieci.*

CARATTERISTICHE DEI CONTATORI

Lancaster[7] ha elencato un certo numero di caratteristiche dei contatori che possono essere così riassunte:

- **Modulo**

  Il *modulo* di un contatore è quello che abbiamo chiamato n, cioé il numero di stati diversi che il contatore attraversa prima di ripetersi. Un contatore binario a quattro bit ha modulo 16; un contatore a decade ha modulo 10; un contatore divisore per 7, ha modulo 7. In un contatore a *modulo variabile*, n può assumere qualsiasi valore all'interno di una gamma di valori.

- **Peso**

Molti contatori della serie di circuiti integrati 7400 sono *contatori pesati*, cioè è possibile assegnare un valore pesato ad ogni uscita dei flip-flop del contatore. Sommando i prodotti dello stato logico per il peso di ogni flip-flop, è possibile calcolare lo *stato* del contatore. Per esempio: Supponiamo che i pesi di un contatore binario siano D=8, C=4, B=2 e A=1; allora l'uscita binaria DCBA=1101 avrà valore 13.

$$13 = 8 \times 1 + 4 \times 1 + 2 \times 0 + 1 \times 1$$

- **Direzione di conteggio**

Molti contatori della serie 7400 sono dei *contatori in avanti* (up-counter), cioè contano solo nella direzione che aumenta la grandezza dello stato calcolato in base a dei pesi. Il contatore 7490 conta da 0 a 9 e poi torna a 0, il 7493 conta da 0 a 15 e poi torna a 0. Invece i contatori 74190, 74192 e 74193 sono copletamente diversi: sono contatori *up/down*, cioè possono contare sia nella direzione che aumenta che in quella che diminuisce la grandezza dello stato. In alcune applicazioni è necessario avere questa possibilità.

- **Sincronismo**

I contatori possono essere asincroni (o ripple) o sincroni. In un contatore asincrono, un cambiamento dell'uscita di un flip-flop può produrre un cambiamento del flip-flop seguente che a sua volta può produrre un cambiamento nel flip-flop seguente e così via. Questi contatori sono poco usati perchè presentano sia dei delay di propagazione che delle uscite spurie dette "glitch". Vengono quindi preferiti dei contatori sincroni i quali cambiano l'uscita dei flip-flop in base all'applicazione di un singolo impulso di clock, allora le uscite cambiano istantaneamente e non si hanno né delay di propagazione né glitch. Mentre i contatori 7490, 7492 e 7493 costano poco ma sono asincroni, i 74160, 74161, 7462, 7463, 74190, 74191 e 74193, costano di più, ma sono sincroni.

- **Possibilità di avere preset**

È importante che un contatore possa essere sia azzerato che *presettato* ad un valore voluto, all'interno dei limiti di conteggio. Il contatore 7490 può essere presettato al valore 0 e 9, mentre il 7493 può essere presettato solo a 0, cioè azzerato. Gli altri contatori che abbiamo elencato possono essere presettati a qualsiasi valore all'interno delle loro possibilità di conteggio.

- **Possiblità di collegamento in cascata (cascadability)**

Alcuni contatori possono essere collegati in sequenza, o in cascata, in modo che l'uscita del primo diventi l'ingresso del secondo, l'uscita del secondo l'ingresso del terzo e così via. Il modulo del contatore che si ottiene é il prodotto dei moduli dei singoli contatori.

Quindi tre contatori a decade in cascata formano un contatore a modulo 1000. Di solito i contatori asincroni scattano quando giunge il fronte negativo dell'impulso di clock, mentre i contatori sincroni di solito scattano quando giunge il fronte positivo dell'impulso di clock.

In genere, in alcuni casi, è meglio avere dei contatori il cui modulo è dato dalla somma dei moduli dei singoli contatori piuttosto che dei contatori il cui modulo è dato dal prodotto dei singoli moduli. Uno di questi casi potrebbe essere quello in cui si desiderasse avere un contatore a modulo 273. I contatori dal 74160 al 74163 e quelli dal 74190 al 74193 sono *collegabili in cascata* e si possono collegare in modo piuttosto strano per dei contatori.

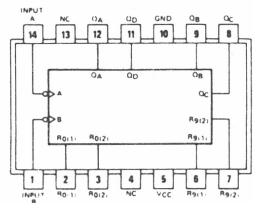
Lancaster ha anche descritto alcuni concetti che servono a progettare i contatori di lunghezza speciale e questi circuiti trattano anche di stati non desiderati, problemi di azzeramento, coincidenze di autoannullamento, impulsi di reset troppo brevi, tempi di salita e discesa dell'impulso di reset lenti e rumori di fondo.

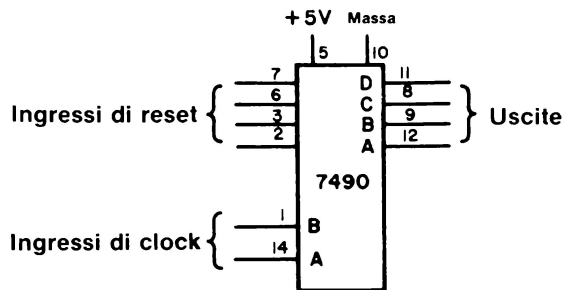
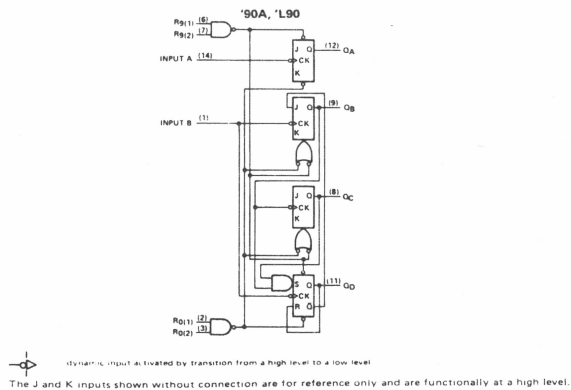
Nella tabella 6 vengono messi a confronto alcuni contatori della serie 7400; in seguito parleremo delle caratteristiche di solo alcuni di essi.

Tabella 6. Contatori della serie 7400.

Chip	Modulo	Direzione	Tipo	Preset?	Possibilità collegamento in cascata?	Organizzazione	Clock
7490	10	up	asincrono	0e9	no	2 x 5	fronte negativo
7492	12	up	asincrono	0	no	2 x 6	fronte negativo
7493	16	up	asincrono	0	no	2 x 8	fronte negativo
74160	10	up	sincrono	si	si	1 x 10	fronte positivo
74161	16	up	sincrono	si	si	1 x 16	fronte positivo
74162	10	up	sincrono	si	si	1 x 10	fronte positivo
74163	16	up	sincrono	si	si	1 x 16	fronte positivo
74176	10	up	asincrono	si	si	2 x 5	fronte negativo
74177	16	up	asincrono	si	si	2 x 8	fronte negativo
74190	10	up/down	sincrono	si	si	1 x 10	fronte positivo
74191	16	up/down	sincrono	si	si	1 x 16	fronte positivo
74192	10	up/down	sincrono	si	si	1 x 10	fronte positivo
74193	16	up/down	sincrono	si	si	1 x 16	fronte positivo
74196	10	up	asincrono	si	si	2 x 5	fronte negativo
74197	16	up	asincrono	si	si	2 x 8	fronte negativo

7490 CONTATORE A DECADI

Il contatore a decadi 7490 ha la seguente configurazione di pin.

7490

e schema a blocchi,

All'interno del circuito integrato c'è l'equivalente di tre flip-flop J-K master-slave, un flip-flop R-S e alcune porte,

Vogliamo sottolineare che questo schema è importante perchè illustra che un contatore è tipicamente composto di flip-flop e porte, messe in modo da produrre il modulo voluto. Altri dettagli sui flip-flop J-K possono essere reperiti nel Bugbook II.

La tabella della verità applicabile al funzionamento del contatore 7490 usato come contatore di decadi è mostrata qui sotto.

BCD COUNT SEQUENCE
(See Note A)

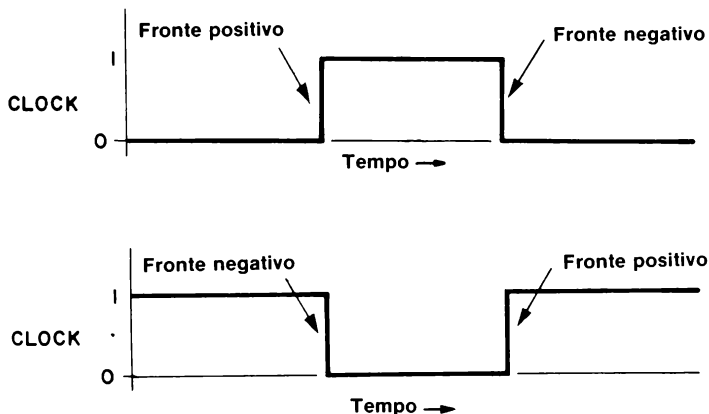
COUNT	OUTPUT			
	Q_D	Q_C	Q_B	Q_A
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	L	H	H	L
7	L	H	H	H
8	H	L	L	L
9	H	L	L	H

'90A, 'L90
RESET/COUNT FUNCTION TABLE

RESET INPUTS				OUTPUT			
$R_{0(1)}$	$R_{0(2)}$	$R_{9(1)}$	$R_{9(2)}$	Q_D	Q_C	Q_B	Q_A
H	H	L	X	L	L	L	L
H	H	X	L	L	L	L	L
X	X	H	H	H	L	L	H
X	L	X	L	COUNT			
L	X	L	X	COUNT			
L	X	X	L	COUNT			
X	L	L	X	COUNT			

FRONTE POSITIVO E NEGATIVO DI UN IMPULSO DI CLOCK

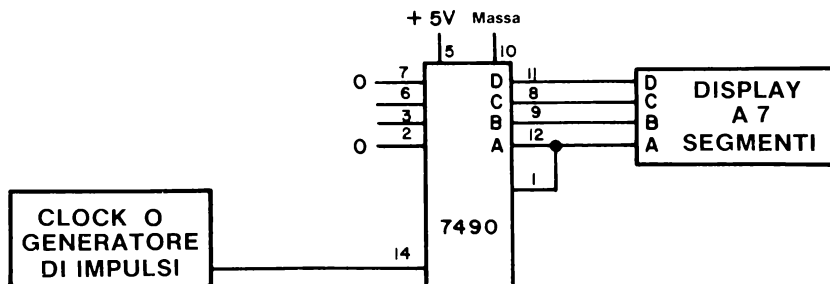
Il contatore a decade 7490 è uno dei più importanti dispositivi elettronici che contengono flip-flop e che cambiano stato sotto il comando di un impulso di clock. Di solito è molto importante saper a quale fronte dell'impulso di clock si determina uno stato di transizione.

Abbiamo già parlato dei fronti positivi e negativi nel capitolo 12, ma li ripetiamo perchè il transitorio ingenerato dal fronte dell'impulso è veramente importante.

Diamo le definizioni dei termini che usiamo più spesso:

Fronte positivo	Passaggio dell'impulso di clock dallo 0 logico all'1 logico.
Fronte negativo	Passaggio dell'impulso di clock dall'1 logico allo 0 logico.
Flip-flop edge triggered	Tipo di flip-flop che necessita di una velocità minima di cambiamento nell'impulso di clock per cambiare il suo stato di uscita.
Triggered	Impulso che fa iniziare una azione. Può anche essere il fronte di un impulso.

FORMA D'ONDA DIGITALE PER UN CONTATORE A DECADE 7490

Il termine *forma d'onda digitale* è stato definito come rappresentazione grafica di un segnale digitale, che quindi esprime le variazioni degli stati logici digitali in funzione del tempo. Chiaramente, in un circuito digitale complesso, sono necessarie diverse forme d'onda per poter capire e rappresentare il comportamento del circuito.

Esaminate il seguente circuito che rappresenta un contatore a decade 7490; vedrete che l'ingresso di clock è collegato all'ingresso A (pin 14) e che Q_A è collegato all'ingresso B (pin 1)

Di seguito è riportata in serie di forme d'onda relative a questo circuito misurate nei seguenti punti (dall'alto in basso).

Impulso di clock A: al pin 14 del chip 7490.

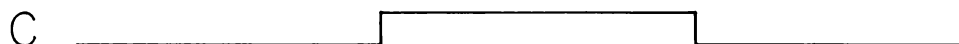
A: Uscita A al pin 12 del chip 7490.

B: Uscita B al pin 9 del chip 7490.

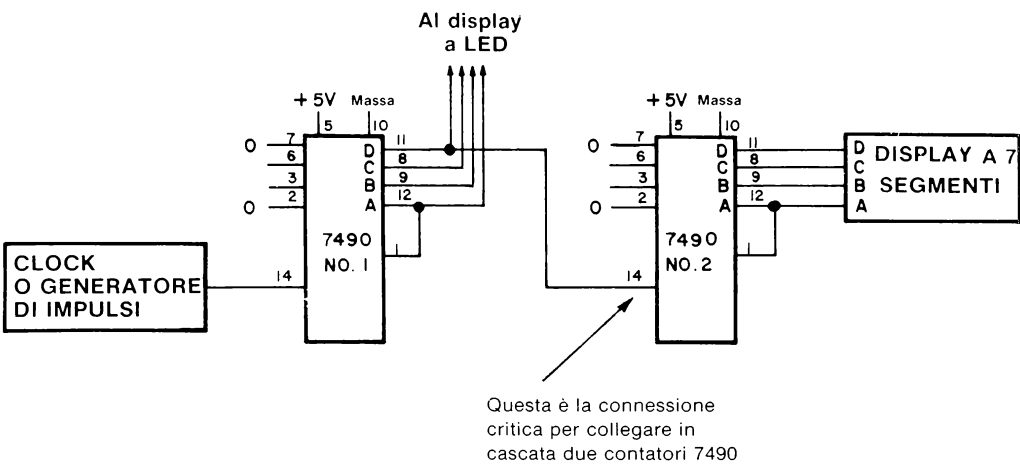
C: Uscita C al pin 8 del chip 7490.

D: Uscita D al pin 11 del chip 7490.

Considerate il passaggio da 3 a 4 del display a sette segmenti; quando il dispaly è a 3 le uscite del contatore 7490 si trovano nei seguenti stati: A=1, B=1, C=0 e D=0. Questi stati logici corrispondono alla parola di quattro bit del binario decimale codificato (BCD) 0011, alla quale corrisponde un display del numero 3. Adesso osservate cosa succede quando giunge, tra 3 e 4, il fronte negativo dell'impulso di clock: Sia A che B tornano allo 0 logico e C passa all'1 logico. La nuova uscita BCD del contatore è 0100 cui corrisponde un display del numero 4.

Si ha un altro passaggio molto importante fra 9 e 0 del display a sette segmenti. Notate che questo passaggio avviene quando giunge il fronte negativo del *singolo impulso di clock all'uscita D.* Ogni volta che arrivano dieci impulsi di clock al pin Ingresso di clock A si verifica un singolo impulso di clock all'uscita D ed è per questo che il contatore a decade 7490 è detto contatore divisore-per-dieci!

COLLEGAMENTO IN CASCATA DEI CONTATORI A DECADE 7490

Collegare in cascata significa collegare due o più circuiti digitali o circuiti integrati simili, in modo che l'uscita di uno diventi l'ingresso del prossimo. Di seguito è illustrato uno schema che mostra due contatori 7490 collegati in cascata. Con un circuito di questo tipo si ottiene un contatore divisore per 100 che può quindi contare da 00 a 99. Il collegamento critico di due contatori in cascata è quello tra l'uscita D del contatore n. 1 e l'ingresso di clock A del contatore n. 2. Quindi l'uscita D del primo contatore 7490 agisce come ingresso di clock del secondo contatore, il quale cambia stato ogni volta che al suo primo ingresso di clock arriva il fronte negativo.

Il numero di collegamenti in cascata può continuare fino al numero di decine volute. Nella Figura 13-1 vengono mostrati quattro contatori a decade in cascata. Abbiamo modificato lo schema in modo che il digit più significativo, il digit delle migliaia, venga a trovarsi all'estrema sinistra e che il digit meno significativo, quello delle unità, venga a trovarsi all'estrema destra. Questo tipo di circuito permette di contare gli impulsi di clock da 0000 a 9999. Collegando una per una le quattro uscite D si ottengono dei contatori divisori per dieci, poi divisori per cento, poi divisori per mille ed infine divisori per diecimila.

Per azzerare simultaneamente tutti e quattro i contatori, bisogna collegare l'uscita clear che si trova al pin 2 di ogni contatore ad una sola sorgente di impulsi di clock clear, che potrebbe essere meccanica, come un generatore di impulsi (come mostrato di seguito), o potrebbe essere generato da un singolo impulso di uscita del microcomputer.

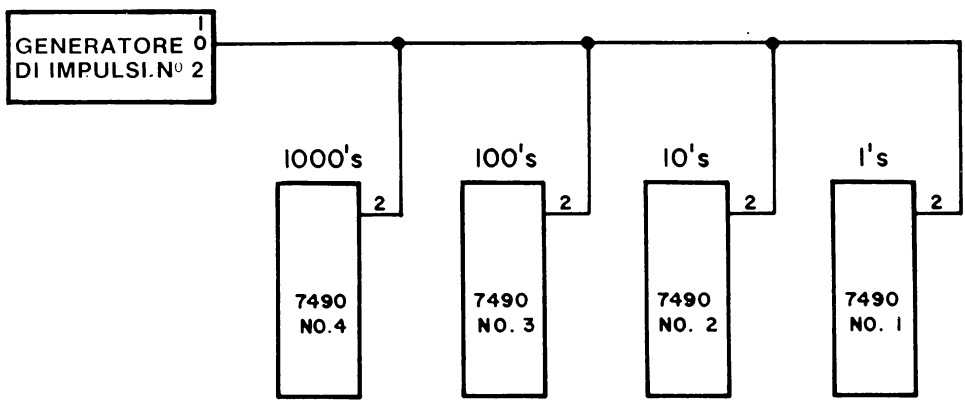
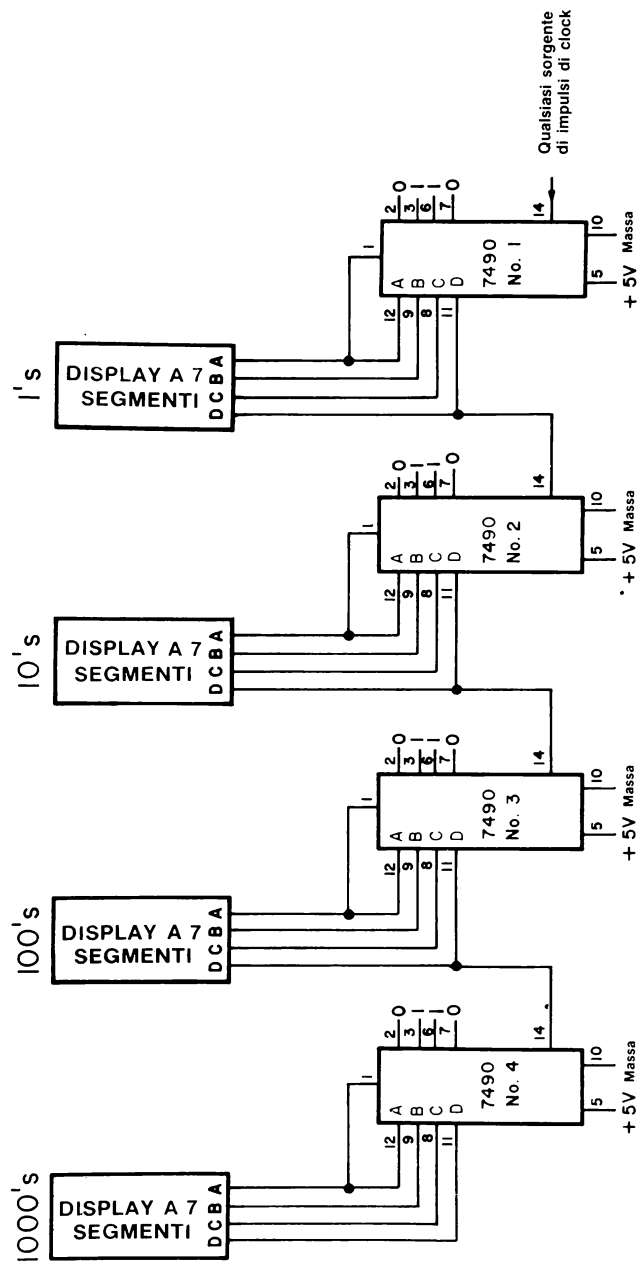



Figura 13-1. Quattro contatori a decade 7490 in cascata, ognuno dei quali è collegato col
segmenti a LED. Con questo circuito si può contare da 0000 a 9999.

7490 CONTATORE BIQUINARIO

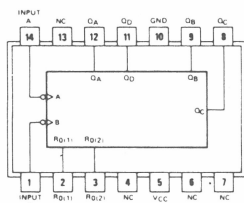
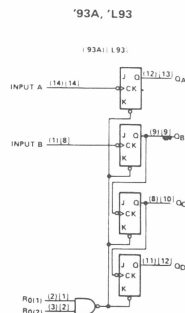
Un *contatore biquinario* (divisore per cinque e divisore per due) è ottenuto applicando il segnale di ingresso di clock all'ingresso B, collegando l'uscita Q_D all'ingresso A e avendo quindi l'uscita divisore per dieci all'uscita Q_A. La tabella della verità per questo tipo di operazione è la seguente:

'90A, 'L90
BI-QUINARY (5-2)
(See Note B)

COUNT	OUTPUT			
	Q_A	Q_D	Q_C	Q_B
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	H	L	L	L
6	H	L	L	H
7	H	L	H	L
8	H	L	H	H
9	H	H	L	L

La possibilità di diventare contatore divisore-per-cinque del 7490 è utile quando si usa il display a sette segmenti a LED di cinque digit della Hewlett-Packard o il display a matrice di punti a LED MAN-2A 5 x 7.

7493 CONTATORE BINARIO

Il contatore binario 7493 contiene quattro flip-flop J-K master-slave messi in modo che l'uscita di un flip-flop dia il clock ai seguenti flip-flop. Per contare normalmente a modulo 16, Q_A è collegato all'ingresso B

La tabella della verità è una semplice progressione di quattro bit binari dall'uscita Q_A all'uscita Q_D e un impulso di reset permette di mettere il contatore allo stato 0.

'92A, '93A, 'L93
RESET/COUNT FUNCTION TABLE

RESET INPUTS		OUTPUT			
$R_{0(1)}$	$R_{0(2)}$	Q_D	Q_C	Q_B	Q_A
H	H	L	L	L	L
L	X	COUNT			
X	L	COUNT			

NOTES: A. Output Q_A is connected to input B for BCD count.
B. Output Q_D is connected to input A for bi-quinary count.
C. Output Q_A is connected to input B.
D. H = high level, L = low level, X = irrelevant

'93A, 'L93
COUNT SEQUENCE
(See Note C)

COUNT	OUTPUT			
	Q_D	Q_C	Q_B	Q_A
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	L	H	H	L
7	L	H	H	H
8	H	L	L	L
9	H	L	L	H
10	H	L	H	L
11	H	L	H	H
12	H	H	L	L
13	H	H	L	H
14	H	H	H	L
15	H	H	H	H

7492 CONTATORE

Nel caso voleste fare un orologio da 60 minuti e aveste quindi bisogno di un contatore divisore-per-60, potreste usare il contatore divisore-per-dodici 7492 che è fatto come un contatore 2 x 6. Qui sotto riportiamo le informazioni che di solito dà la Texas Instruments Incorporated:

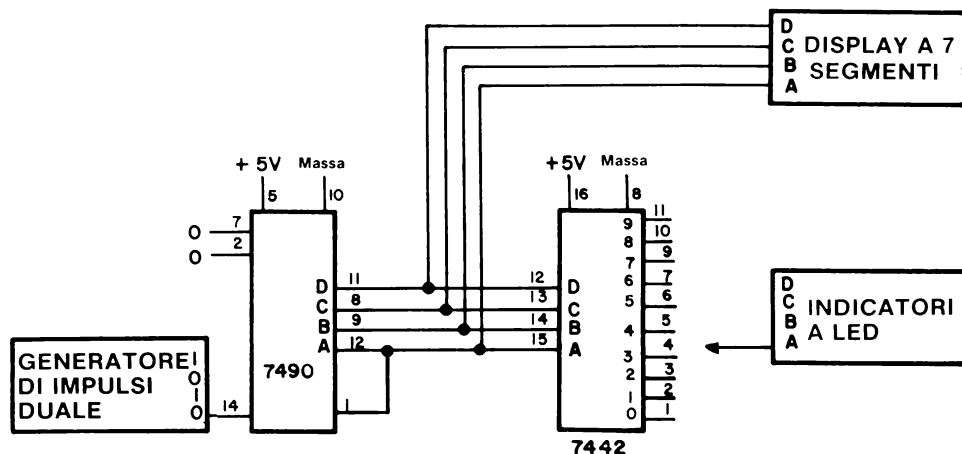
'92A
COUNT SEQUENCE
(See Note C)

COUNT	OUTPUT			
	Q_D	Q_C	Q_B	Q_A
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	H	L	L	L
7	H	L	L	H
8	H	L	H	L
9	H	L	H	H
10	H	H	L	L
11	H	H	L	H

Osservate che nei contatori 7490, 7492 e 7493 c'è sempre un singolo contatore a modulo 2, cioè un flip-flop.

GLITCH DIGITALI

Nel capitolo 11 abbiamo già parlato del concetto di race condition, nella quale due segnali indipendenti dal tempo "si rincorrono" l'un l'altro e provocano una azione o un gruppo di azioni. Questa condizione, che è spesso provocata da carenze di programmazione e/o da delay di propagazione, può produrre un impulso non voluto detto "glitch". Nel capitolo 11 abbiamo già fatto un esempio di glitch e adesso ne facciamo un altro.

Considerate il seguente circuito generatore di sequenze nel quale un contatore a decade asincrono 7490 è collegato ad un decodificatore 7442:

Quando questo circuito lavora ad altre frequenze possono apparire degli impulsi digitali - glitch - sui canali di uscita numerati dal decodificatore, come viene mostrato nel diagramma temporale successivo. I glitch sono causati dalla azione asincrona di Q_A che cambia prima che l'ingresso B del contatore possa rispondere. Succede che per un istante - il tempo del glitch - l'uscita del decodificatore non è determinata. La durata del glitch dipende dal delay di propagazione e dai tempi di commutazione dei flip-flop che stanno all'interno del contatore 7490.

Di solito il problema dei glitch non sussiste per quei decodificatori che vengono usati per generare impulsi di selezione di dispositivi di I/O (vedere capitolo 17) perchè vengono usati dei segnali di sincronizzazione per fornire uno strobe alle operazioni del decodificatore dopo che si siano stabilizzate tutte le linee di indirizzamento. Una buona progettazione logica tiene conto dei tempi di assestamento dei segnali logici: il microprocessor è stato progettato tenendo conto di queste cose.

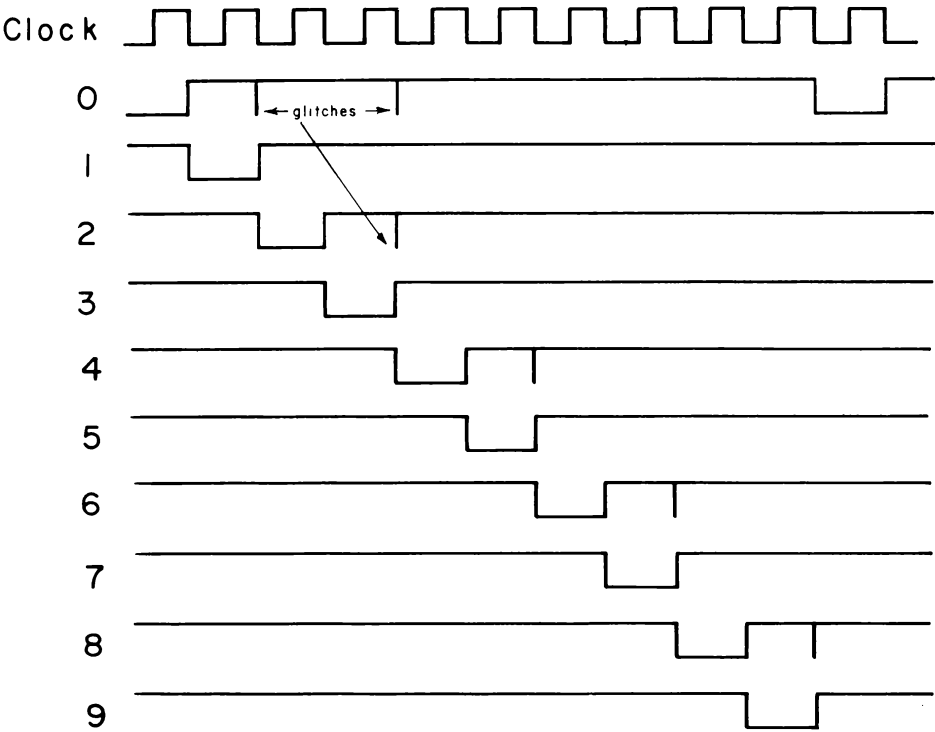



Figura 13-2. Diagrammi temporali che illustrano la creazione di alcuni "glitch" nel circuito 7490/7442 mostrato precedentemente. Questi "glitch" appaiono sui canali pari del decodificatore.

INTRODUZIONE AGLI ESPERIMENTI

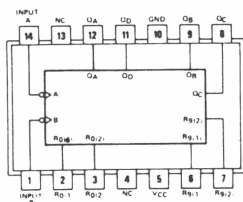
I seguenti esperimenti mostrano le caratteristiche proprie dei contatori usati come dispositivi misuratori di frequenza e come generatori di sequenze.

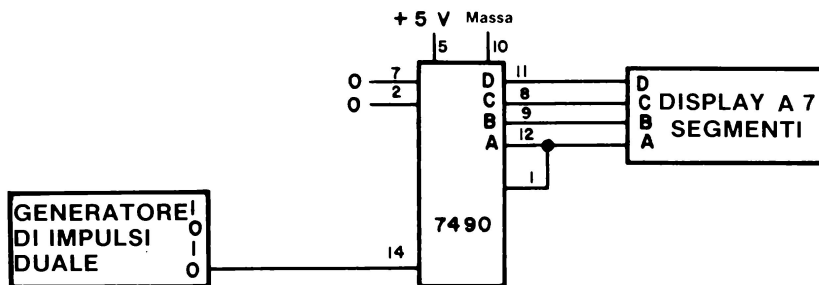
Esperimento N.	Commento
1	Mostrare il funzionamento del chip 7490 come contatore a decade.
2	Misurare la frequenza del clock dell'Outboard LR-5 o LR-25.
3	Mostrare come si possa agganciare e bloccare l'uscita di un contatore, mediante l'uso di un latch 7475.
4	Mostrare l'uso del latch/display della Hewlett-Packard HP 5082-7300.
5	Mostrare il funzionamento di un circuito generatore di sequenze a decade.
6	Mostrare il funzionamento di un circuito generatore di sequenze esadecimale.
7	Mostrare come si costruisce un contatore a più decadi e come lo si usa come misuratore di frequenze.
8	Mostrare l'uso dell'Outboard latch/display a tre digit LR-28.
9	Mostrare come si possa eliminare un rimbalzo di contatto di uno switch meccanico NAND a 2-ingressi.
10	Mostrare come si possa usare un microcomputer come generatore di sequenze ottali.

ESPERIMENTO N. 1

Scopo

Lo scopo di questo esperimento è di mostrare il funzionamento del circuito integrato 7490 usato come contatore o decade. La sorgente di impulsi di clock è un generatore di impulsi meccanico dotato di logica antirimbalzo.

Configurazione dei pin del circuito integrato

7490

Schema del circuito

Passo 1

Collegate il circuito e alimentate il breadboard. Uno degli errori più frequenti nel collegamento di questo tipo di circuito è quello di dimenticare il collegamento tra il pin 1 e il pin 12.

Passo 2

Che numero appare sul display a sette segmenti? Premete e rilasciate il generatore di impulsi. Che cosa succede?

Noi abbiamo visto che il display si è incrementato quando abbiamo rilasciato il generatore di impulsi. Non vi diciamo il valore del contatore apparso sul nostro display perchè ogni chip 7490 non appena lo si alimenta comincia a contare da un valore imprecisato.

Passo 3

Premete e rilasciate il generatore di impulsi quindici-venti volte. Cosa vedete?

Vediamo che il display a sette segmenti si incrementa ogni volta che rilasciamo il generatore di impulsi. I valori che abbiamo visto sono compresi tra 0 e 9 e dopo il valore 9 viene lo 0.

Passo 4

Premete e rilasciate il generatore di impulsi fino a che non appare sul display il numero 4. Adesso staccate il collegamento tra il pin 2 e la massa e collegatelo all'1 logico. Cosa succede?

Appena tolto il collegamento a massa, è apparso 0 sul display.

Passo 5

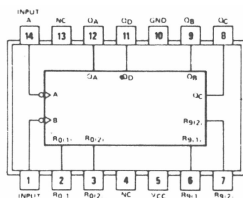
Ricollegate il pin 2 a massa e staccate il collegamento tra il pin 67 e la massa. Cosa succede?

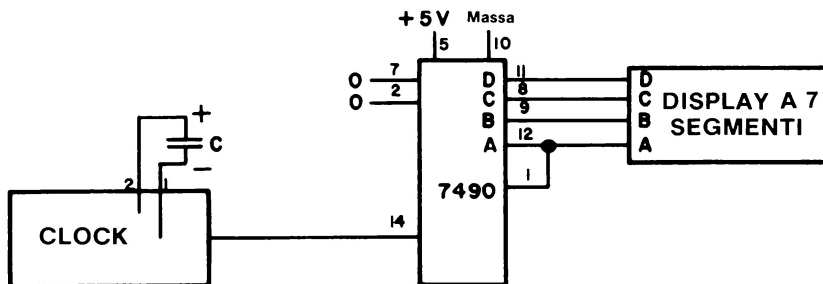
Appena tolto il collegamento a massa del pin 7 è apparso 9 sul display, cioè abbiamo resettato il contatore al valore 9. Nel passo precedente invece lo avevamo resettato al valore 0, o azzerato.

ESPERIMENTO N. 2

Scopo

Lo scopo di questo esperimento è di misurare la frequenza del clock degli Outboard LR-5 e LR-25 mediante l'uso di in contatore a decadi 7490.

Configurazione dei pin del circuito

7490

Schema del circuito

Passo 1

Collegate il circuito e alimentate il breadboard. Se vedete apparire sul display a sette segmenti solo il numero 8, vuol dire che il clock lavora ad una frequenza troppo alta e allora dovete diminuirla usando un grosso condensatore che, se di 0,33μF, dovrebbe portarla a circa 1 Hz.

Passo 2

Con un orologio determinate quanti impulsi di clock arrivano ogni minuto e scrivete la risposta negli spazi sotto.

Con un condensatore di ...... μF si hanno .... impulsi di clock al minuto.

Passo 3

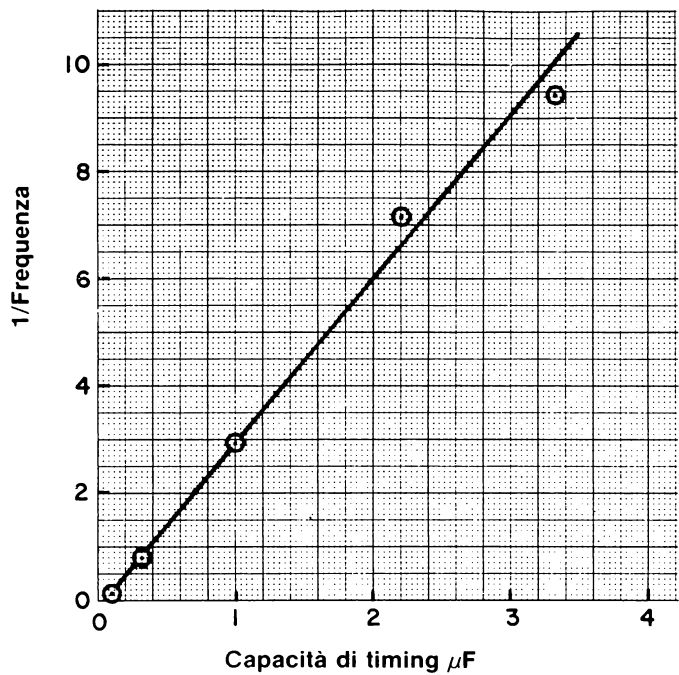
Adesso cambiate il valore della capacità del condensatore inserito nei due pin dell'Outboard di clock. Determinate il numero di impulsi di clock *al secondo* per ogni valore di capacità e mettete i risultati trovati nella tabella qui sotto:

Capacità di Timing in μF	Numero di impulsi di clock al secondo

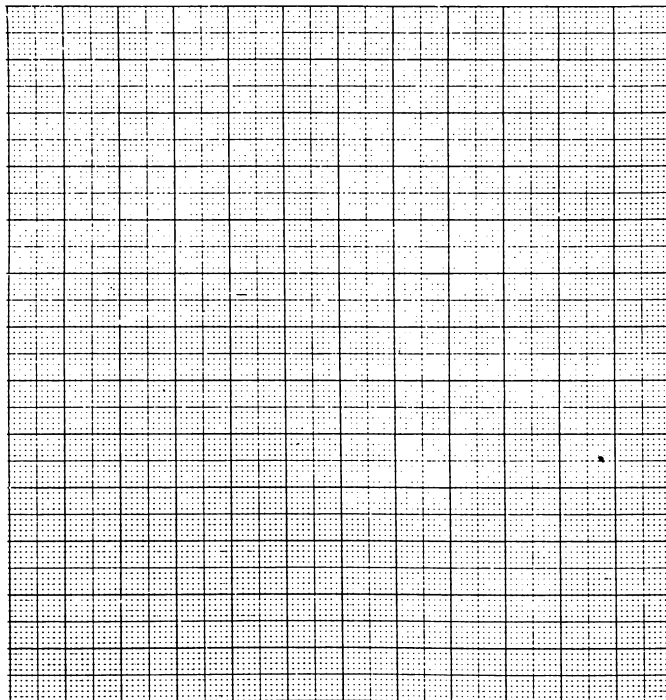
Se usate un condensatore elettrolitico, assicuratevi che il terminale negativo sia collegato al socket di massa dell'Outboard di clock.

Quando noi abbiamo eseguito le rilevazioni, abbiamo ottenuto i seguenti risultati:

Capacità di Timing in μF	Frequenza (conteggi/s)	1/Frequenza (s./conteggio)
0,10	6,97	0,144
0,10	7,35	0,136
0,33	1,23	0,81
1,0	0,34	2,9
2,2	0,140	7,2
3,3	0,106	9,4

Se si mette su un diagramma cartesiano l'inverso della frequenza e il valore della capacità si ottiene una funzione lineare, come mostrato di seguito. Osservate che la tolleranza del condensatore non superava ±20%.

Passo 4

Calcolate i reciproci della frequenza in funzione dei valori della capacità e mettete i risultati nel diagramma della pagina seguente. La funzione ottenuta è lineare? Cosa rappresenta la pendenza della curva?

Capacità di Timing in μF	Frequenza (conteggi/s)	1/Frequenza (s/conteggio)

In base alle nostre misurazioni abbiamo calcolato una pendenza di 3,08 Hz^{-1} μF^{-1}. quindi, per valori abbastanza grandi di capacità, la relazione tra frequenze e capacità è circa,

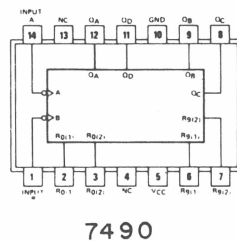
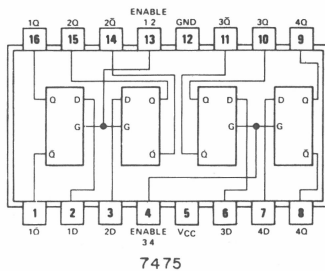
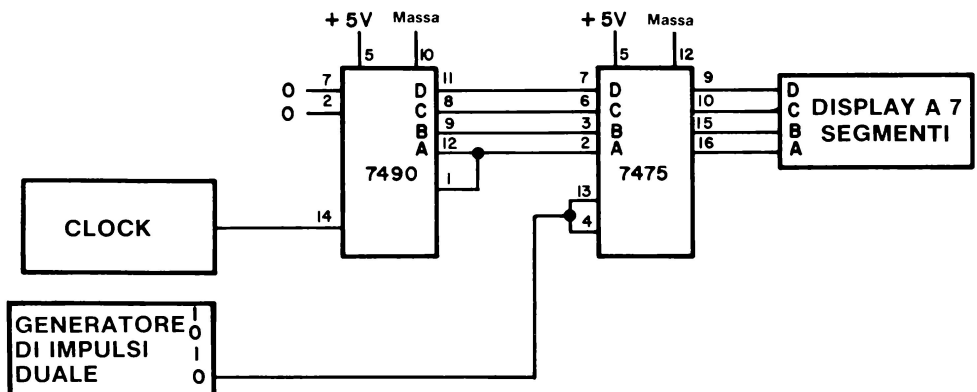
$$\nu = 0{,}32/C$$

dove la frequenza ν è espressa in Hz e la capacità in microfarad.

ESPERIMENTO N. 3

Scopo

Lo scopo di questo esperimento è di dimostrare come si può agganciare l'uscita di un contatore a decadi 7490.

Configurazione dei pin del circuito integrato

Schema del circuito

Passo 1

Collegate il circuito e alimentate il breadboard. Potete determinare il comportamento del circuito se ricordate l'esperimento col chip 7475 del capitolo 11.

Passo 2

La frequenza di clock deve essere di circa 1 Hz quindi usate un condensatore di 0,33 μF collegato con l'Outboard di clock.

Premete il generatore di impulsi e tenetelo premuto; dovreste vedere dei cambiamenti sul display ogni volta che il contatore 7490 "conta" gli impulsi di clock che arrivano al pin 14. Adesso lasciate andare il generatore di impulsi. Che numero appare sul display? Perchè non cambia più?

Non cambia più perchè, una volta che è stato rilasciato il generatore di impulsi, il latch 7475 è disabilitato e quindi impedisce ad ogni nuovo valore di conteggio di arrivare al display. In altre parole, il dato presente, quando è giunto al fronte negativo del segnale del generatore di impulsi, è stato agganciato e bloccato. Cioè il 7475 funziona come memoria di quattro bit "ricorda" la parola BCD di quattro bit che era presente immediatamente prima che giungesse il fronte negativo dell'impulso in ingresso.

Passo 3

Usando il generatore di impulsi, bloccate sul display il numero decimale 7. Se adesso lasciate andare il generatore di impulsi il numero 7 resta bloccato sul display?

Si, ci rimane fino a che non viene premuto un'altra volta il generatore di impulsi.

Passo 4

Togliete il condensatore dall'Outboard di clock; adesso la frequenza dovrebbe essere compresa tra i 90 e i 100 kHz. Premete il generatore di impulsi e notate che il 7490 sta contando tanto velocemente che sul display sembra apparire il numero 8 mentre in realtà si tratta dei sette segmenti che si accendono uno dopo l'altro. Rilasciate il generatore di impulsi; dovreste bloccare uno dei dieci digit decimali. L'azione di aggancio e bloccaggio avviene quando giunge il fronte negativo dell'impulso di clock ai pin 4 e 13 del chip 7475.

Premete e rilasciate alcune volte il generatore di impulsi, dovreste vedere ogni volta dei numeri decimali diversi perchè casuali.

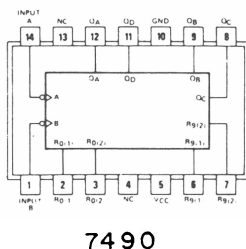
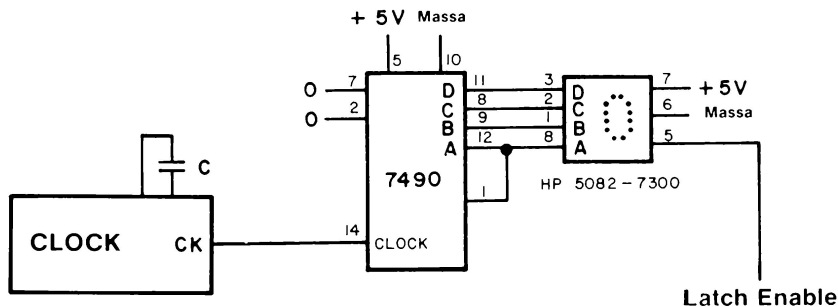
Di solito nei microcomputer vengono usati dei latch per prendere e memorizzare dei dati che cambiano con estrema rapidità. Mediante una accurata scelta dell'ingresso ai pin 4 e 13 è possibile agganciare e bloccare l'esatta informazione che interessa. Questo principio è molto importante perchè permette, quando si interfaccia il microcomputer, di agganciare e bloccare dei dati anche se presenti solo per un breve periodo di tempo.

ESPERIMENTO N. 4

Scopo

Lo scopo di questo esperimento è di mostrare l'uso del latch/display HP 5082-7300 della Hewlett-Packard per agganciare e bloccare e poi visualizzare l'uscita di un contatore a decade 7490.

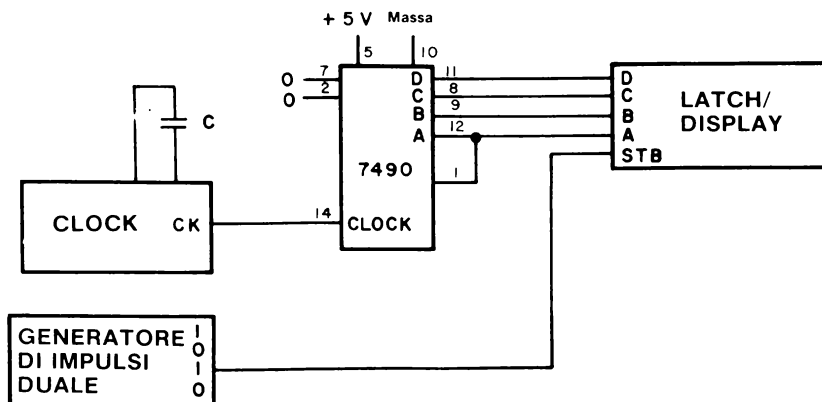
Configurazione dei pin del circuito integrato.

7490

Schema del circuito

Passo 1

Questo esperimento è simile all'Esperimento n. 3 nel quale però venivano usati separatamente un latch - il chip 7475 - e un display - il display a sette segmenti -. L'HP 5082-7300 è un prodotto molto sofisticato della tecnologia digitale; contiene al suo interno un latch a quattro bit, un decodificatore/pilota e un gruppo di LED disposti in modo da poter visualizzare i digit da 0 a 9. La differenza tra questo latch/display e la combinazione del chip 7475 col display a sette segmenti è che il *latch/display viene abilitato da uno stato logico 0, mentre il chip 7475 viene abilitato da uno stato logico 1.*

Detto in altro modo, uno 0 logico applicato all'ingresso STROBE (pin 5) dell'HP 5082-7300, permette di visualizzare il conteggio che è in uscita dal chip 7490. Nel prossimo capitolo discuteremo i termini *abilitare, disabilitare* e *fornire uno strobe*.

Passo 2

Collegate il circuito. Se avete un Outboard latch/display LR-29 o un Outboard LR-25, potete sostituire con uno dei due l'HP 5082-7300 e collegare invece questo circuito:

Passo 3

Alimentate il breadboard e mettete l'ingresso STB del latch/display (pin 5) allo stato logico 0. Usate un condensatore di una capacità che vi permetta di avere dall'Outboard di clock una frequenza di 1 Hz. Cosa vedete sul latch/display?

Noi vediamo l'uscita del contatore 7490, cioè una sequenza di numeri decimali.

Passo 4

Adesso mettete l'ingresso STB del latch/display all'1 logico. Il conteggio visibile sul display?

Si. Infatti avete disabilitato il latch e memorizzato l'ultimo digit decimale che era stato visualizzato prima che il latch fosse disabilitato.

Passo 5

Togliete dall'Outboard di clock il condensatore aumentando quindi la frequenza di clock, che adesso è di circa 100 kHz. Se state usando l'Outboard LR-25 o LR-29, premete e rilasciate il generatore di impulsi illustrato nello schema. Se invece state usando il latch/display, collegate un generatore di impulsi all'ingresso Latch-Enable. Collegate l'uscita 1 del generatore di impulsi al pin 5 del latch/display poi premete e rilasciate il generatore di impulsi. Cosa vedete?

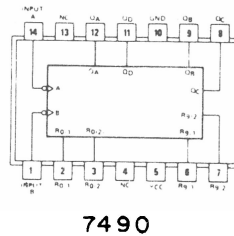
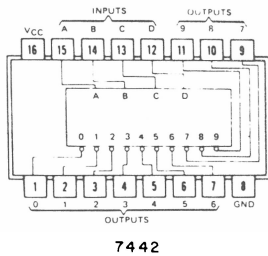
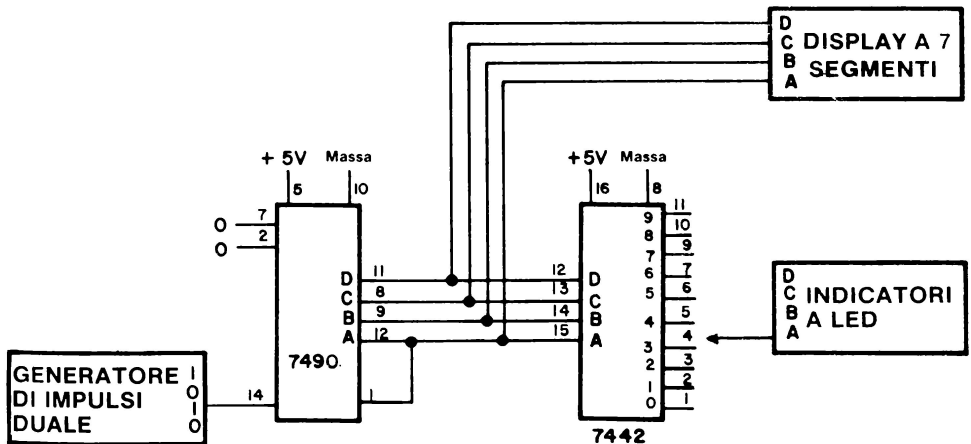
Quando il generatore di impulsi è premuto, si vede sul display un'uscita data dalla somma di tutti i singoli display di conteggio del 7490 che si susseguono con ritmo rapidissimo. Quando rilasciate il generatore di impulsi, agganciate e bloccate uno dei digit tra 0 e 9.

Premete e rilasciate alcune volte il generatore di impulsi e osservate che ogni volta agganciate e bloccate uno specifico digit tra 0 e 9. Userete tre di questi latch/display per agganciare e bloccare l'informazione presente sul data bus bidirezionale a otto bit del vostro microcomputer MMD-1. In un capitolo successivo parleremo del modo con cui si agganciano e bloccano dei dati dal data bus bidirezionale.

ESPERIMENTO N. 5

Scopo

Lo scopo di questo esperimento è di mostrare il funzionamento del decodificatore 7442 usato come generatore di sequenza a decade.

Configurazione dei pin dei circuiti integrati

Schema del circuito

Passo 1

Collegate il circuito illustrato nello schema. Dovreste ricordare alcuni degli esperimenti che avete fatto col decodificatore 7442 nel capitolo 12.

Passo 2

Alimentate il breadboard e collegate gli indicatori a LED al canale 0 (pin 1) del decodificatore 7442. Premete e rilasciate più volte il generatore di impulsi fino a che l'indicatore a LED non si spegne. A questo punto qual'è il numero che appare sul display a LED?

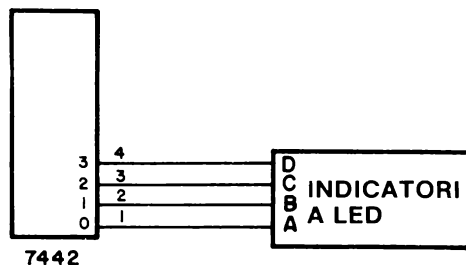
L'indicatore a LED si spegne quando sul display a sette segmenti appare il numero 0.

Passo 3

Collegate adesso l'indicatore a LED al canale 7 del pin 9. Percorrerete ancora una volta il ciclo di conteggio e fermatevi quando l'indicatore si spegne. Che numero decimale appare adesso sul display?

Quando l'indicatore si spegne vediamo il decimale 7.

Passo 4

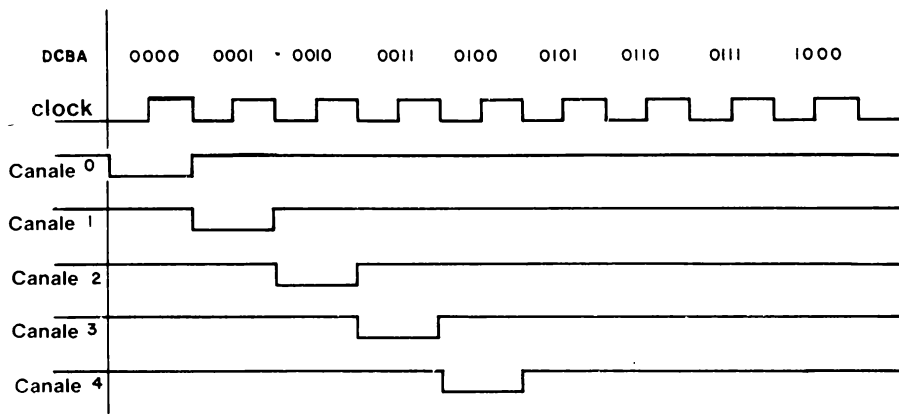
Collegate i quattro indicatori a LED dal canale 0 al canale 3 del decodificatore 7442, nel modo illustrato qui sotto:

Premete e rilasciate più volte il generatore di impulsi. Cosa vedete sui quattro indicatori a LED?

Abbiamo visto che gli indicatori si spengono in sequenza, prima quello del canale 0, poi il canale 1, il canale 2 e infine il canale 3. In ogni istante, quindi, solo un indicatore è spento oppure sono tutti accesi.

Passo 5

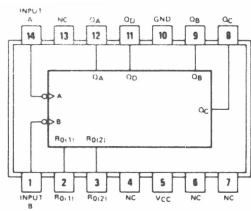
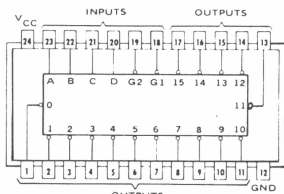
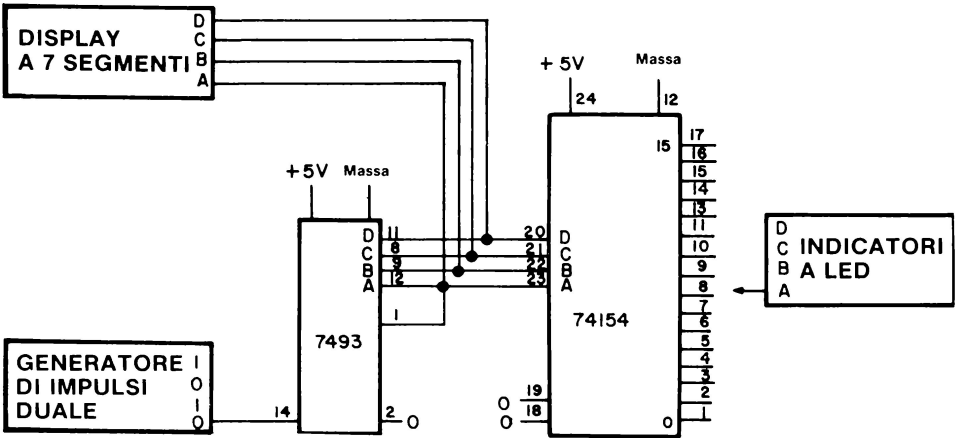
Se riuscite a collegare dieci indicatori a LED ai canali del decodificatore 7442, cosa vedreste premendo e rilasciando più volte il generatore di impulsi in ingresso al contatore a decade 7490?

Dovreste vedere che uno stato logico 0 passa attraverso tutti i canali. Solo un canale alla volta ha il corrispondente indicatore spento e l'ordine è quello dei canali, canale 0, canale 1, canale 2 . . . . . . canale 9, canale 0, etc.

Se avete eseguito questo esperimento, avrete costruito un generatore di sequenza a decade, infatti avete visto lo stato logico 0 passare in sequenza tutti i dieci canali di uscita, come mostrato di seguito per i canali da 0 a 4:

ESPERIMENTO N. 6

Scopo

Lo scopo di questo esperimento è di mostrare il funzionamento del decodificatore 74154 usato come generatore di sequenze esadecimali.

Configurazione dei pin dei circuiti integrati

Schema del circuito

Passo 1

Collegate il circuito. Notate che nel chip 74154 i canali di uscita sono all'interno nello schema e che i numeri dei pin sono invece all'esterno. Nell'Esperimento n. 3 del capitolo 12 avete già trovato la tabella della verità del decodificatore 74154.

Passo 2

Alimentate il breadboard e azzerate il contatore 74493, o mettendo momentaneamente il pin 2 del contatore binario all'1 logico o facendo ciclare il contatore fino a che non appaia uno 0 sul display a sette segmenti.

Controllate lo stato logico del canale 0 (pin 1) del decodificatore 74154. È allo 0 logico?

Dovrebbe esserlo. Se non lo è, c'è qualcosa di sbagliato o nel vostro circuito o nel circuito integrato. Assicuratevi che i pin 18 e 19 del decodificatore siano entrambi allo 0 logico.

Passo 3

Premete e rilasciate per una volta il generatore di impulsi in modo che sul display a sette segmenti appaia il numero decimale 1. Controllate lostato logico del canale 1 (pin 2) del decodificatore 74154. È allo 0 logico?

Si. Tutti gli altri sono all'1 logico?

Si.

Passo 4

Premete e rilasciate per una volta il generatore di impulsi in modo che sul display a sette segmenti appaia il numero decimale 1. Controllate lo stato logico del canale 1 (pin 2) del decodificatore 74154. È allo 0 logico?

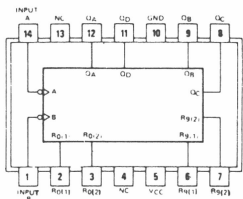
Le risposte sono entrambe si. A questo punto dovreste aver già capito che state passando in sequenza tutti i canali di uscita del decodificatore 74154 uno alla volta. Sul display a sette segmenti appare il numero del canale che in quel momento è allo 0 logico, mentre tutti gli altri sono all'1 logico.

Continuate a passere in sequenza tutti i canali di uscita del decodificatore e ogni volta controllate lo stato del canale che compare sul display a sette segmenti.

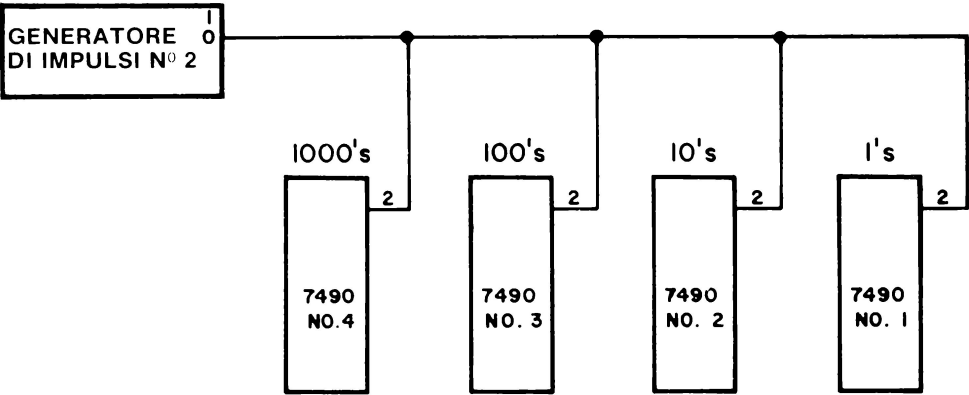
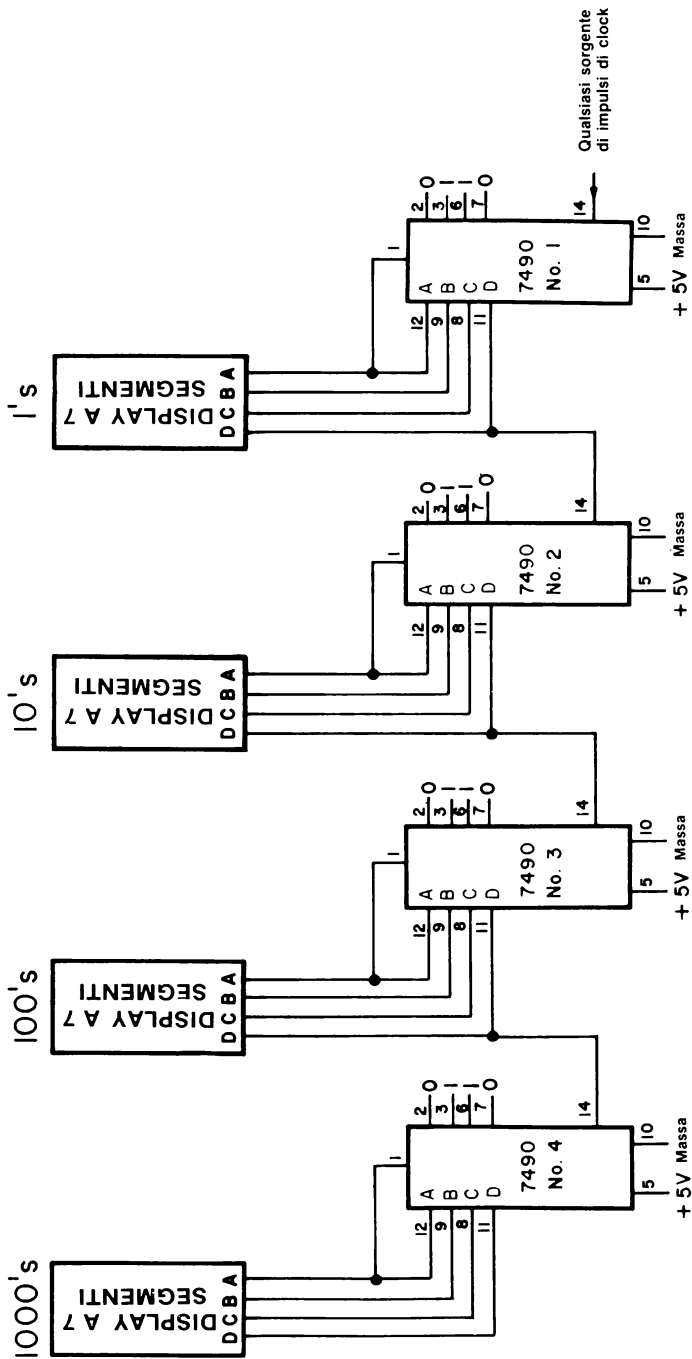
ESPERIMENTO N. 7

Scopo

Lo scopo di questo esperimento è di costituire un contatore a due, tre o quattro decadi e di usarlo per misurare una larga banda di frequenza.

Configurazione dei pin del circuito integrato

7490

Schema del circuito

Nel seguito è riportato lo schema di un contatore a quattro decadi. Se volete azzerare simultaneamente tutti e quattro i contatori dovete collegare il pin 2 di ognuno di essi ad un generatore di impulsi come mostrato qui sotto:

Passo 1

Collegate il circuito. Dovreste riuscire a farlo stare su un'unica piastra, ma siccome lo scopo di questo esperimento è di mostrare come funziona un contatore a più decadi, se non ci stesse, basta un contatore a due decadi, o meglio, a tre decadi. Se state lavorando in gruppo ogni componente del gruppo potrebbe costruire un contatore a due decadi su una piastra separata. Poi potreste collegare tra loro le due breadboard formando il circuito di un contatore a quattro decadi. Alimentate i breadboard con un'unica sorgente di +5V e *conservate il circuito che avete costruito per l'Esperimento n. 9.*

Passo 2

Questo esperimento è simile all'Esperimento n. 2 di questo capitolo. Misurerete la frequenza associata a diversi condensatori. Collegate l'ingresso di clock al pin 14 del primo contatore 7490 (il contatore delle unità). Aiutandovi con un orologio da polso con cronometro, misurate il numero di conteggi in un dato intervallo di tempo, per esempio un minuto. Per far partire il conteggio premete e rilasciate il generatore di impulsi che azzera il contatore e poi mettete i risultati ottenuti in questa tabella.

Capacità di Timing in μF	Conteggi minuto	Frequenza conteggi/s	Frequenza^{-1} s/conteggio	Frequenza x Capacità 10^{-8} Farad/s
nessuna				
0,00001				
0,00002				
0,00005				
0,0001				
0,0002				
0,0005				
0,001				
0,002				
0,005				
0,01				
0,02				
0,05				
0,1				
0,22				
0,47				
1,0				
2,2				

Se la frequenza del clock è tropppo alta, saltate la misura corrispondente. Se avete un contatore a due decadi, dovreste riuscire a misurare le frequenze associate a capacità di 0,01μF. Con tre decadi dovreste arrivare fino a 0,001μF. Con quattro decadi dovreste riuscire a misurare frequenze associate con capacità di 0,0001μF e anche meno.

Per fare la seguente tabella abbiamo usato un contatore a sei decadi:

Capacità di Timing in μF	Conteggi/ minuto	Frequenza conteggi/s	Frequenza^{-1} s/conteggio	Frequenza x Capacità 10^{-8} Farad/s
nessuna	--	72300	0,0000138	
0,00001	--	33300	0,0000300	33,3
0,00002	--	23300	0,000043	46,5
0,00005	--	11900	0,000084	59,5
0,0001	--	6540	0,000153	65,4
0,0002	--	3510	0,000285	70,2
0,0005	--	1540	0,00065	76,9
0,001	--	766	0,00131	76,6
0,005	--	141	0,00708	70,7
0,01	--	73,5	0,0136	73,5
0,02	--	39,6	0,0253	79,2
0,05	--	17,7	0,0565	88,5
0,1	400	6,67	0,15	66,6
0,33	74		0,81	40,7
1	19,5		3,08	32,5
2,2	8,38		7,16	30,7
3,3	6,36		9,43	35,0
0,33	75		0,80	41,6

Il prodotto della frequenza per la capacità dovrebbe essere una costante quando la capacità è inferiore a 0,0005μF e nel Bugbook I, nel capitolo 4, abbiamo sperimentato questa relazione con un modello di Outboard di clock più vecchio. In questo momento però ci interessa il motivo per il quale il prodotto tra capacità e frequenza risulta, in questo esperimento, un po' troppo basso, ma lo abbiamo riportato ugualmente per fare un esempio di come, spesso, i valori di capacità indicati siano sbagliati. Noi pensiamo che i valori indicati per la capacità sopra i 0,1μF siano affidabili, ma alcune volte la tolleranza diventa del 50%!

Possiamo calcolare la capacità interna dei Timer 555 presenti nell'Outboard di clock, dividendo il prodotto frequenza-capacità per la frequenza che si ha senza condensatore, 72300 Hz. Quindi

$$\text{Capacità interna} = \frac{76 \times 10^{-8}\ \text{Farad/s}}{72{,}300\ \text{s}^{-1}} = 10{,}5\ \text{pF}$$

Questa capacità interna deve essere sommata alla capacità esterna per ottenere la capacità totale, con la quale è possibile calcolare la frequenza del timer. Allora con una capacità esterna di 10 pF la capacità totale è di 20,5 pF e la frequenza diventa

$$\text{Frequenza} = \frac{76 \times 10^{-8}\ \text{Farad/s}}{20{,}5 \times 10^{-12}\ \text{Farad}} = 37{,}100\ \text{Hz}$$

ESPERIMENTO N. 8

Scopo

Lo scopo di questo esperimento è di costruire un contatore a tre decadi usando tre chip 7490 e l'Outboard latch/display a tre digit LR-28.

Configurazione dei pin del circuito integrato

7490

Schema del circuito

7490

7490

7490

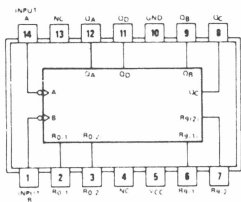
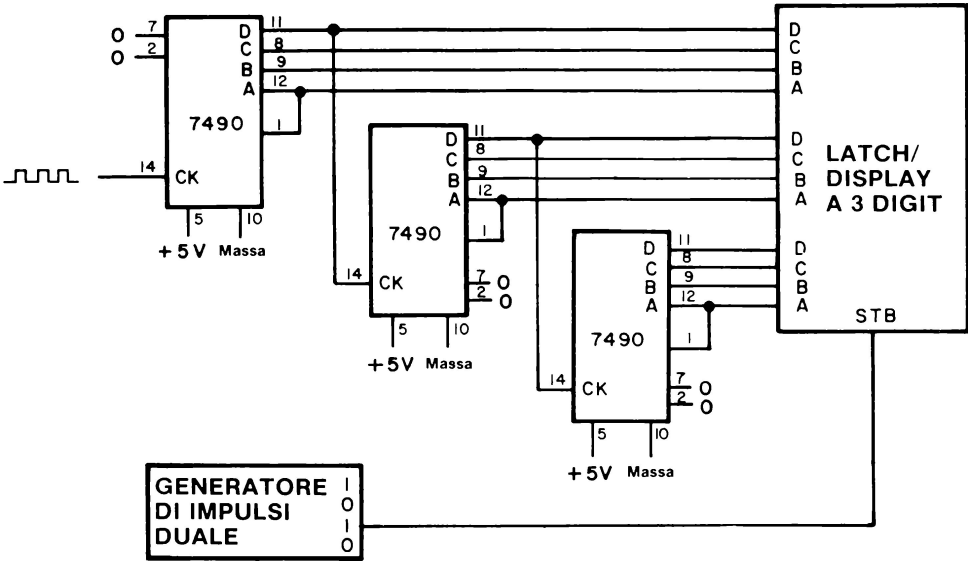
+5V Massa

+5V Massa

+5V Massa

LATCH/
DISPLAY
A 3 DIGIT

STB

GENERATORE
DI IMPULSI
DUALE

Passo 1

Collegate il circuito e ripetete l'esperimento proposto all'Esperimento n. 7, mettendo i risultati nella seguente tabella. Eseguite le misure con più condensatori.

Capacità di Timing in μF	Conteggi/ minuto	Frequenza conteggi/s	Frequenza $^{-1}$, s/conteggio	Frequenza x Capacità, 10^{-8} Farad/s
0,0001				
0,0002				
0,0005				
0,001				
0,002				
0,005				
0,01				
0,02				
0,05				
0,1				
0,22				
0,47				
1				
2,2				
4,7				

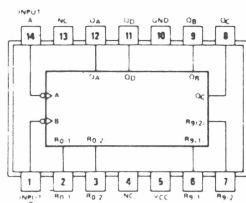
Potreste fare il contatore riutilizzando il circuito dell'Esperimento n. 7.

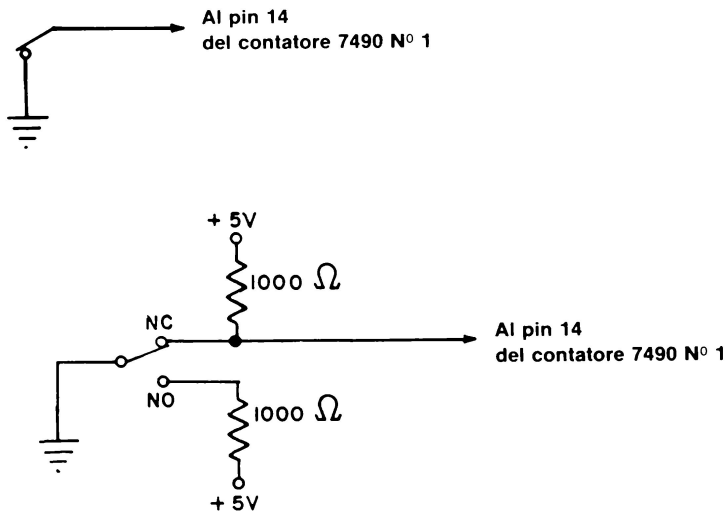
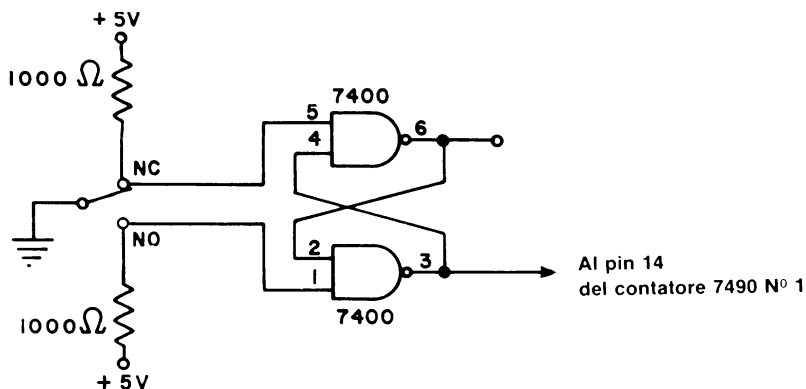
Se non avete ancora collegato il circuito dell'Esperimento n. 7, conservate questo circuito e usatelo per l'Esperimento n. 9.

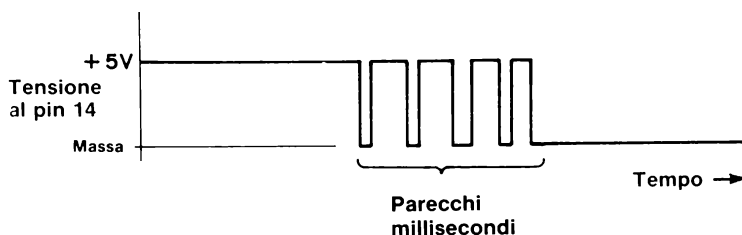
ESPERIMENTO N. 9

Scopo

Lo scopo di questo esperimento è di mostrare come nasce il fenomeno del rimbalzo da contatto e come lo si possa eliminare. Collegherete alcuni circuiti molto semplici che producono uno o più rimbalzi.

Configurazione dei pin del circuito integrato

7490

Schemi dei circuiti

Passo 1

Il fenomeno del rimbalzo da contatto può essere schematicamente rappresentato col seguente diagramma temporale:

Lo stato logico oscilla tra lo 0 logico e l'1 logico fino a che, finalmente, si stabilizza, in questo caso, allo 0 logico. Se tentate di mettere a massa un filo inizialmente a +5V, potreste osservare un fenomeno di questo tipo se disponeste di un oscilloscopio con memoria.

Passo 2

Il vostro contatore a più decadi deve essere collegato, che sia il circuito dell'Esperimento n. 7 o quello dell'Esperimento n. 8. Non vi occorrono più di tre decadi. Collegate il chip 7490 in modo da poter azzerare tutti i contatori contemporaneamente.

Passo 3

Collegate un filo lungo all'ingresso del contatore a più decadi e mettetelo subito a massa. Quanti rimbalzi avete visto? Azzerate il contatore e staccate il filo dalla massa. Avete visto altri rimbalzi?

Passo 4

Ripetete il processo "porre a massa/azzeramento/staccare dalla massa" fino a quando vi siete convinti che è difficile generare uno e un solo impulso di clock mettendo meccanicamente a massa un filo.

Passo 5

Collegate il secondo degli ultimi circuiti presentati usando lo stesso switch SPDT o alcuni fili e resistori che simulino lo switch SPDT.

Fate funzionare lo switch SPDT o il suo equivalente e dimostrate che c'è ancora un rimbalzo di contatto. Alcuni switch sono più "rimbalzanti" di altri, noi ne abbiamo trovato uno che è rimbalzato 563 volte prima di fermarsi allo stato logico 0.

Passo 6

Adesso collegate il terzo circuito che consiste in un paio di porte NAND collegate come semplice latch e, tra l'altro, abbiamo già dato una descrizione di questo circuito nel capitolo 11. Fate funzionare lo switch dotato di logica antirimbalzo e dimostrate che produce un solo impulso ogni volta che viene premuto e rilasciato. Provate alcune volte; ogni volta viene generato un singolo impulso.

Discussione

Un altro modo di dotare di logica antirimbalzo uno switch è di (a) determinare quando viene fatto il primo contatto, (b) attendere un predeterminato periodo di tempo, circa 10 ms, e (c) controllare se il contatto continua o no. Un'attesa di 10 ms è sufficiente per permettere allo switch "rimbalzante" di assestarsi.

Il microcomputer MMD-1 usa una tecnica di questo tipo. Il delay di 10 ms viene generato all'interno della KEX, la EPROM che governa il funzionamento del microcomputer. Il programma funziona esattamente come abbiamo già detto: (a) determina quando viene premuto un tasto, (b) entra nel loop di delay di 10 ms, (c) controlla se il tasto è ancora premuto e (d) va ad eseguire una subroutine che adempie alla funzione corrispondente al tasto premuto.

ESPERIMENTO N. 10

Scopo

Lo scopo di questo esperimento è di mostrare come si possa usare il microcomputer come generatore di sequenze ottali.

Programma

Indirizzo di memoria LO	Byte di istruzione	Codice mnemonico	Descrizione
271	076	MVI A	Muovi in modo immediato il byte seguente nell'accumulatore.
272	376	<B2>	Contenuto iniziale dell'accumulatore.
273	323	OUT	Metti in uscita il contenuto dell'accumulatore.
274	000	000	Codice di dispositivo della porta 0.
275	016	MVI C	Muovi in modo immediato il byte seguente nel registro C.
276	144	<B2>	Byte di timing per il loop di delay.
277	315	CALL	Chiama la subroutine che si trova all'indirizzo di memoria contenuto nei prossimi due byte.
300	277	-	Indirizzo di memoria LO.
301	000	-	Indirizzo di memoria HI.
302	015	DCR C	Decrementa di 1 il contenuto del registro C.
303	302	JNZ	Se il contenuto del registro C è 000, ignora questa istruzione; altrimenti salta all'indirizzo di memoria contenuto nei prossimi due byte.
304	277	-	Indirizzo di memoria LO.
305	003	-	Indirizzo di memoria HI.
306	323	OUT	Metti in uscita il contenuto dell'accumulatore.
307	000	000	Codice di dispositivo della porta 0.
310	007	RLC	Ruota, di una posizione verso sinistra, il contenuto dell'accumulatore. Il bit meno significativo ed il carry flag sono entrambi al valore del bit che è stato posto in uscita dalla posizione del bit più significativo.

311	303	JMP	Salta all'indirizzo di memoria contenuto nei prossimi due bytes.
312	275	-	Indirizzo di memoria LO.
313	003	-	Indirizzo di memoria HI.

Passo 1

Caricate ed eseguite sul microcomputer il programma che abbiamo riportato. Cosa vedete sulla porta 0?

Noi vediamo che sette degli otto indicatori sono accesi e che l'indicatore spento si sposta in sequenza passando da una all'altra delle otto posizioni. La velocità di questo spostamento è di circa una posizione di bit al secondo.

Passo 2

Confrontate l'uscita della porta 0 con quella del decodificatore nell'Esperimento n. 5. Il fenomeno base è il medesimo?

Si. In entrambi i casi, uno stato logico 0 passa in sequenza attraverso un certo numero di canali di uscita; quindi, il programma che avete caricato, forza il microcomputer a comportarsi come un generatore di sequenze ottali.

Discussione

Il programma richiama una subroutine che si trova agli indirizzi HI=000 e LO=277; è una subroutine che permette di ottenere un delay di tempo di 10 ms ed è contenuto nella KEX; è la stessa subroutine che viene usata per dotare di logica antirimbalzo i tasti della tastiera. Ogni volta che avete bisogno di un delay di 10 ms, o multiplo di 10 ms, potrete richiamare questa subroutine, sarà sempre a quell'indirizzo.

DOMANDE RIEPILOGATIVE

Le seguenti domande servono ad aiutarvi a ripassare ciò che avete visto in questo capitolo.

1. Per convertire la frequenza di 50 Hz in un'uscita digitale di un ciclo al secondo, bisogna dividere la frequenza digitale originale per 60. Riuscite a proporre un contatore in cascata che possa risolvere questo problema.

2. Cos'è un glitch? Quali dei seguenti contatori possono aver dei glitch nei conteggi in uscita: 74160, 74163, 7492 e 7490?

3. Se voleste misurare una frequenza digitale di 10 MHz, avreste bisogno di un contatore a quante decadi? Motivate la risposta.

4. Quali cambiamenti bisogna apportare al programma dell'Esperimento n. 10 per aumentare o diminuire la velocità della sequenza?

RISPOSTE

1. Per costruire un contatore a modulo 60 si possono seguire due metodi: (a) usare un 7490 come contatore a modulo 5 e un 7492 come contatore a modulo 12; oppure (b) usare un 7490 come contatore a modulo 10 e un 7492 come contatore a modulo 6.

2. Un "glitch" è un impulso non voluto o uno stato logico causato, per esempio, da un race condition prodottasi in un circuito digitale.

3. La risposta non è unica. Se volete una precisione al settimo digit dovete usare almeno sette contatori a decade, ma se vi basta una precisione minore, vi bastano cinque o sei contatori a decade che però richiedono un gating pulse molto più accurato, tra 10 e 100 ms. Nel prossimo capitolo imparerete altre cose sui gating pulse. Se questo impulso è necessario, come può essere un generatore d'impulsi, avete bisogno di un contatore a nove o dieci decadi per avere una precisione appena sufficiente.

4. Potete aumentare o diminuire la velocità della sequenza cambiando il delay di tempo del programma. Il modo più semplice è quello di cambiare il byte di timing che si trova all'indirizzo LO=276. Un byte del valore 001 produce un delay di 10 ms tra un'uscita e l'altra, mentre un byte di valore 377 produce un delay di 2,55 s.

CAPITOLO 14

GATING DI SEGNALI DIGITALI

INTRODUZIONE

Questo capitolo introdurrà al concetto di *gating* cioè ai principi di utilizzo di chip a porte, utilizzati per controllare il passaggio delle informazioni digitali da un dispositivo digitale ad un altro. La vostra attenzione verrà centrata sul *gated counter*, contatore in cui una porta controlla gli impulsi che devono essere contati. Proverete il funzionamento dei chip 7400, 7408 e 7432 usati come dispositivi di gating.

OBIETTIVI

Alla fine di questo capitolo sarete in grado di:

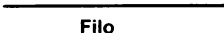
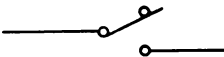
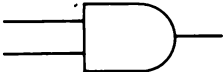
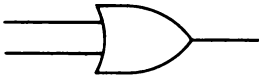
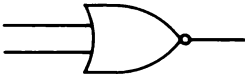
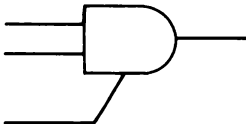
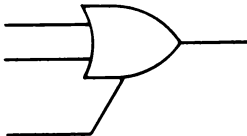
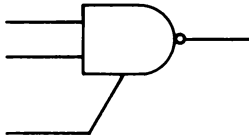
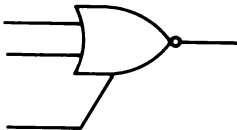
- Elencare gli stati logici richiesti per abilitare e disabilitare le porte AND, OR, NAND e NOR.
- Elencare quanti ingressi di gating ci sono nei diversi tipi di porte a più ingressi.
- Elencare diversi tipi di operazioni che possono essere fatte sui segnali digitali.
- Sapere la differenza tra l'uso di una porta come dispositivo logico e come dispositivo di gating.
- Mostrare come si possono usare le tabelle della verità per prevedere il comportamento delle porte e degli elementi di gating.
- Dare la definizione di porta o gate.
- Identificare gli ingressi di gating dei più complessi circuiti integrati MSI e LSI.

COSA È UN SEGNALE DIGITALE?

I *segnali digitali* sono dei segnali discreti o discontinui i cui diversi stati sono degli intervalli discreti e separati. Il segnale digitale tipico dell'elettronica digitale è il *segnale binario* che viene definito come una tensione o una corrente che porta una informazione sotto forma di cambiamento tra due stati diversi separati da un intervallo discreto. Uno di questi stati è detto stato logico 0 e l'altro è detto stato logico 1. Di solito, per i segnali di tensione, lo stato logico 0 corrisponde al potenziaie di massa mentre lo stato logico 1 corrisponde ad una tensione tra i +3 e i +5V. Per i segnali di corrente, lo stato logico 0 corrisponde ad assenza di corrente (0 mA), mentre lo stato logico 1 di solito corrisponde, a 20 mA.

CHE OPERAZIONI SI POSSONO FARE SU UN SEGNALE DIGITALE?

Prendete in esame la seguente figura,

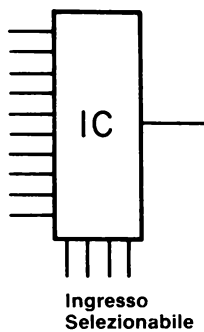
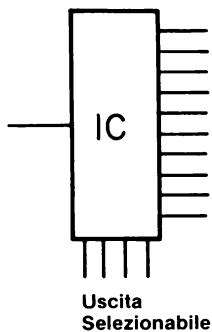
che rappresenta un dispositivo digitale che converte un segnale digitale di ingresso in un segnale digitale di uscita. Supponete che il segnale di ingresso sia un treno di impulsi di clock,

Sorge spontanea una domanda: cosa succede a questo treno di impulsi quando passa in differenti dispositivi digitali? Nella tabella 7 rispondiamo a questa domanda elencando, per i dispositivi digitali più rappresentativi,

- Il dispositivo digitale.
- Il simbolo del dispositivo digitale.
- Le operazioni che il dispositivo digitale esegue sul segnale digitale in ingresso.

L'elenco non pretende di essere completo; esistono infatti altri dispositivi digitali che sommano, sottraggono, confrontano effettuano shift e memorizzano i segnali digitali.

Tabella 7. Alcune delle operazioni più significative tra quelle che possono essere eseguite su un segnale digitale. Non vengono considerate le operazioni che coinvolgono le memorie.

Dispositivo digitale	Simbolo	Operazione
cavo	Filo	Trasmette il segnale senza modificarlo. Ritarda il segnale di 3 ns/m.
invertitore		Inverte il segnale.
driver		Aumenta la corrente di segnale o il livello di tensione
buffer		Aumenta la corrente di segnale o il livello di tensione.
switch SPDT		Trasmette o impedisce di trasmettere un segnale dalla linea di uscita.
porta AND		Trasmette o blocca il segnale.
porta NAND		Trasmette o blocca il segnale.

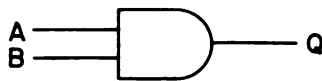
Dispositivo digitale	Simbolo	Operazione
porta OR		Trasmette o blocca il segnale.
porta NOR		Trasmette o blocca il segnale.
porta AND Three-state		Trasmette o blocca il segnale, o disconnette elettronicamente la porta dalla linea del bus di uscita.
porta OR Three-state		Trasmette o blocca il segnale, o disconnette elettronicamente la porta dalla linea del bus di uscita.
porta NAND Three-state		Trasmette o blocca il segnale, o disconnette elettronicamente la porta dalla linea del bus di uscita.
porta NOR Three-state		Trasmette o blocca il segnale, o disconnette elettronicamente la porta dalla linea del bus di uscita.

Dispositivo digitale	Simbolo	Operazione
selezionatore di dati, multiplexer	IC Ingresso Selezionabile	Indirizza o blocca il segnale.
demultiplexer	IC Uscita Selezionabile	Indirizza o blocca il segnale.

PORTE COME DISPOSITIVI LOGICI E COME DISPOSITIVI DI GATING

Le porte hanno due usi fondamentali: come elementi di decisione (o di logica) e come elementi di gating. I due diversi usi delle porte possono essere distinti nel seguente modo:

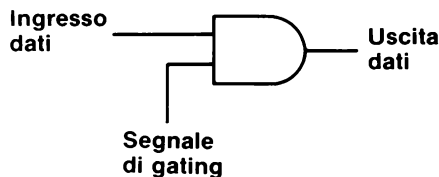
Porta (dispositivo logico) — Circuito con due o più ingressi e un'uscita, la quale dipende da una combinazione logica dei segnali in ingresso. Le quattro porte base sono: AND, OR, NAND e NOR.

Porta (dispositivo di gating) — Circuito con due o più ingressi e un'uscita. Si può identificare un ingresso come ingresso dati mentre gli altri sono ingressi di gating. Lo stato logico dell'ingresso di gating determina se il dato in ingresso deve apparire o meno in uscita.

Quindi si può usare lo stesso dispositivo hardware per due diverse applicazioni. È molto diffuso usare una porta per bloccare o trasmettere un dato.

La tabella della verità è la seguente:

Ingressi		Uscita	
B	A	Q	
0	0	0	
0	1	0	
1	0	0	
1	1	1	(stato unico)

Questa tabella della verità riassume l'uso della porta AND come elemento decisionale. L'uscita Q dipende dalla combinazione logica degli stati logici dei due ingressi A e B.

Prendiamo adesso in esame il seguente schema di una normale porta AND a 2-ingressi,

I due ingressi della porta vengono adesso detti “ingresso dati (input data)” e “segnale di gating (gating signal)” o l’uscita della porta è detta “uscita dati (output data)”. Nonostante questa nuova nomenclatura, la tabella della verità della porta rimane sempre la stessa.

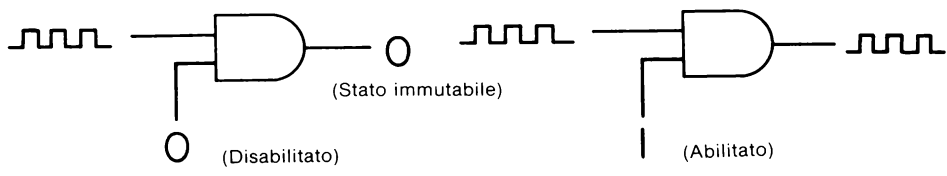
segnale di gating	ingresso dati	uscita dati	
0	0	0	
0	1	0	
1	0	0	
1	1	1	(stato unico)

Questa tabella della verità può essere semplificata nel seguente modo,

segnale di gating	ingresso dati	uscita dati
0	0 o 1	0
1	0	0
1	1	1

Oppure può essere scritta in termini di cosa accade ad un treno di tre impulsi di clock che la attraversa.

	segnale di gating	ingresso dati	uscita dati	
Non abilitato:	0	⎍⎍⎍	0	(stato bloccato)
Abilitato:	1	⎍⎍⎍	⎍⎍⎍	

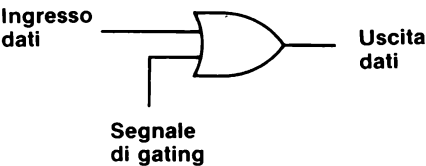
Che può essere rappresentato anche nel seguente modo,

Abbiamo cambiato l’uso di una porta AND da elemento di decisione a elemento di gating. Quando il segnale di gating è allo 0 logico l’uscita della porta AND rimane allo 0 logico; quando il segnale di gating è all’1 logico, l’uscita della porta è allo stesso stato logico dell’altro ingresso.

Quindi è possibile controllare il passaggio dall’ingresso all’uscita di una porta AND di una informazione digitale scegliendo mediante lo stato dell’ingresso di gating. Quanto abbiamo appena detto può essere fatto con una qualsiasi delle quattro porte base: AND, OR, NAND e NOR e adesso lo dimostreremo.

PORTA OR COME ELEMENTO DI GATING

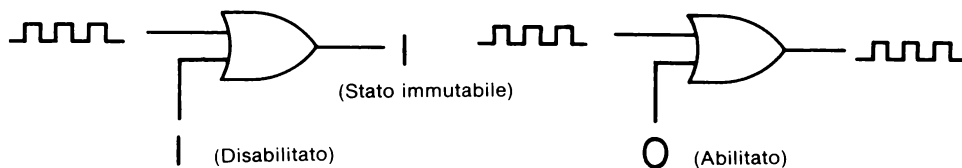
La tabella della verità di una porta OR a 2-ingressi,

Ingressi B	Ingressi A	Uscita Q	
0	0	0	(stato unico)
0	1	1	
1	0	1	
1	1	1	

può essere ridefinita per corrispondere al seguente schema per la stessa porta OR.

segnale di gating	ingresso dati	uscita dati
0	0	0
0	1	1
1	0 o 1	1

Questa tabella della verità può essere rappresentata in termini di cosa accade a un treno di tre impulsi di clock,

	segnale di gating	ingresso dati	uscita dati	
Abilitato:	0			
Non abilitato:	1		1	(stato bloccato)

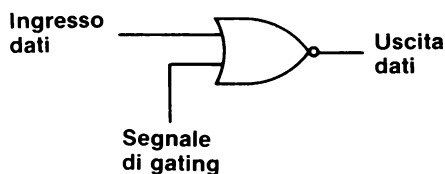
Che può essere schematicamente rappresentato come segue:

Osservate che il segnale di gating all'1 logico blocca il passaggio del treno di impulsi di clock attraverso la porta.

PORTA NOR COME ELEMENTO DI GATING

Il comportamento della porta NOR a 2-ingressi,

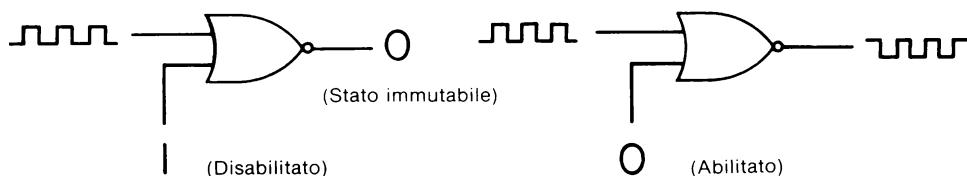
è simile a quello della porta OR a 2-ingressi che abbiamo già discusso. La sua tabella della verità

Ingressi		Uscita	
B	A	Q	
0	0	1	(stato unico)
0	1	0	
1	0	0	
1	1	0	

può essere ridefinita in modo da corrispondere al seguente schema:

segnale di gating	ingresso dati	uscita dati
0	0	1
0	1	0
1	0 o 1	0

Questa tabella della verità può essere rappresentata in termini di cosa accade ad un treno di impulsi di clock,

	segnale di gating	ingresso dati	uscita dati	
Abilitato:	0	⊓⊔⊓⊔	⊔⊓⊔⊓	
Non abilitato:	1	⊓⊔⊓⊔	0	(stato bloccato)

che può essere a sua volta rappresentato dal seguente schema:

Notate che il dato in uscita della porta NOR è il complemento, o l'inverso, del dato in ingresso, quindi una porta NOR, non solo può bloccare il dato in ingresso, ma lo inverte quando il segnale di gating è allo 0 logico.

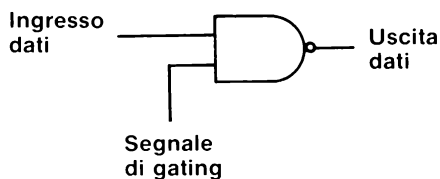
PORTA NAND COME ELEMENTO DI GATING

La porta NAND a 2-ingressi,

si comporta in modo del tutto simile alla porta AND a 2-ingressi. Il dato in ingresso, invece di apparire immutato in uscita, viene invertito da un segnale di gating all'1 logico e questo fatto può essere illustrato nel seguente modo.

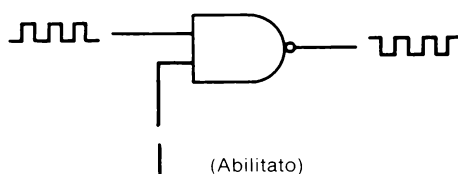
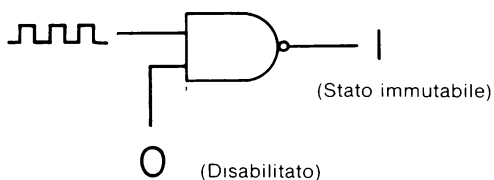
La tabella della verità di una porta NAND a 2-ingressi.

Ingressi		Uscita	
B	A	Q	
0	0	1	
0	1	1	
1	0	1	
1	1	0	(stato unico)

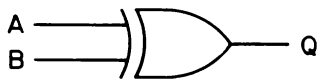
può essere ridefinita in modo da corrispondere alla porta quando si comporta come elemento di gating,

segnale di gating	ingresso dati	uscita dati
0	0 o 1	1
1	0	1
1	1	0

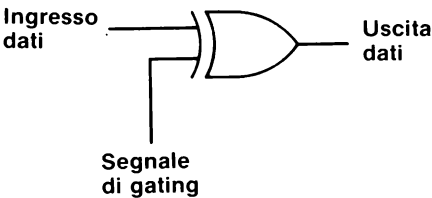
Questa tabella della verità può essere rappresentata in termini di cosa accade ad un treno di impulsi di clock,

	segnale di gating	ingresso dati	uscita dati	
Non abilitata:	0		1	(stato bloccato)
Abilitata:	1			

che viene riassunto nei seguenti due schemi,

Quando il segnale di gating è all'1 logico , il dato in ingresso appare invertito all'uscita della porta.

PORTA OR-ESCLUSIVO COME INVERTITORE CONTROLLABILE

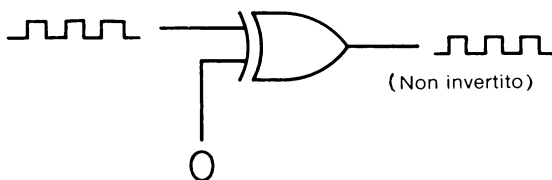
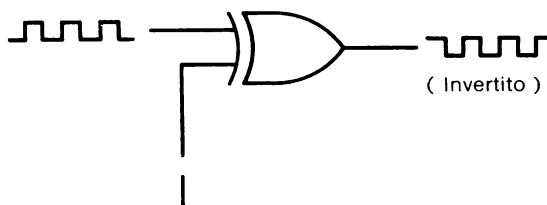
Siccome la porta OR-Esclusivo non ha uno stato di uscita unico,

Ingressi		Uscita
B	A	Q
0	0	0
0	1	1
1	0	1
1	1	0

quando viene usata come elemento di gating si comporta diversamente dalle altre quattro porte, come si può vedere dal seguente schema:

La tabella della verità modificata è la seguente:

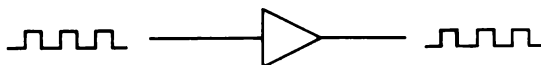
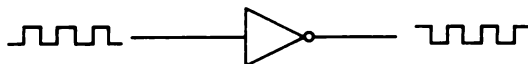
segnale di gating	ingresso dati	uscita dati
0	0	0
0	1	1
1	0	1
1	1	0

cioè, in termini di segnale in ingresso - un treno di impulsi di clock -,

segnale di gating	ingresso dati	uscita dati	
0			(non invertito)
1			(invertito)

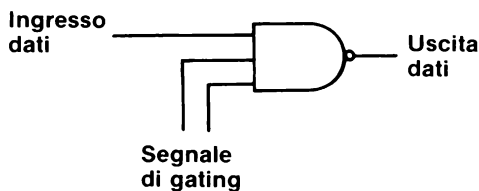
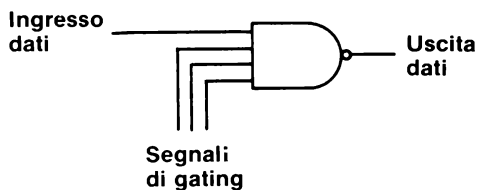
Quindi, con un segnale di gating allo 0 logico, la porta OR-Esclusivo funziona come un semplice buffer,

mentre con un segnale di gating all'1 logico la porta OR-Esclusivo inverte il dato in ingresso,

La porta OR-Esclusivo è detta anche complementatore controllabile o invertitore controllabile perchè ha la possibilità di controllare come trattare il segnale in ingresso. Bisogna però dire che quando viene usata in questo modo non è realmente una porta perchè non ha un'uscita unica.

INVERTITORI, BUFFER E DRIVER

Gli invertitori, i buffer e i driver, come un semplice cavo conduttore, hanno in comune la caratteristica di non bloccare il passaggio di un dato digitale. È possibile modificare i buffer e gli invertitori in modo che possano bloccare il passaggio dei dati, ma ne parleremo nel capitolo che tratta delle tecniche di "three-state bussing".
Il seguente schema è applicabile a questi dispositivi:

Tuttavia i buffer e i driver possono aumentare sia il livello di corrente che di tensione del segnale digitale, ma non possono controllare se il segnale sia presente o no.

È un fatto ribadito più volte che non è facile trasmettere a grande distanza un segnale logico TTL, cioè un potenziale di +5V o di massa. Si può dire che la distanza massima di trasmissione non supera i venti o trenta centimetri (un piede circa) dal chip TTL 7400, infatti potrebbero esserci dei disturbi generati da lampade fluorescenti, motori o altre macchine, o accoppiamento di segnali (capacitivo o induttivo) da un circuito all'altro con la conseguenza di diminuire di molto il livello di affidabilità della trasmissione TTL.

SEGNALI DI GATING MULTIPLI

In questo capitolo abbiamo già visto che un segnale di gating può determinare se un segnale digitale può passare o meno attraverso una semplice porta a 2-ingressi. Come potreste aspettarvi, ci sono delle variazioni su questo tema, per esempio, una porta NAND a 3-ingressi può essere vista nel seguente modo:

cioè con un solo ingresso dati e due ingressi per segnali di gating i quali devono essere entrambi all'1 logico affinchè il dato in ingresso appaia sull'uscita della porta. Se uno dei due segnali di gating è allo 0 logico, il dato in ingresso viene bloccato. Per una porta AND a 4-ingressi, come quella mostrata di seguito,

ci sono tre segnali di gating, i quali devono essere tutti all'1 logico perchè il dato in ingresso passi in uscita.

IL NOME: PORTA (GATE)

Il termine *porta* (in inglese gate) ha un certo numero di significati e quindi è meglio elencare tutte le possibili definizioni che se ne possono dare:

Porta (gate)

1. Circuito con due o più ingressi e un'uscita che dipende dalla combinazione degli stati logici in ingresso. Ci sono quattro porte base, dette AND, OR, NAND e NOR, ma la più usata nei computer è la porta AND.
2. Segnale usato per permettere il passaggio in un circuito di un altro segnale. Il segnale è detto, in molti casi, segnale di gating.
3. Uno degli elettrodi di un transistor ad effetto di campo (FET).
4. Circuito in cui un segnale (di solito un'onda quadra o un impulso) mette un altro segnale in on o off.

Nell'elettronica digitale si incontrano comunemente tutti questi usi e definizioni della parola porta (gate), e sarete voi, che, quando leggete un articolo, dovete capire in che senso viene usato il termine. In questo testo di solito facciamo riferimento alla definizione 1. Le definizioni che abbiamo dato sono prese dal "Modern Dictionary of Electronics" [Rudolf F. Gray, Howard W. Sams & Co., Inc, Indianapolis.]

I VERBI: TO GATE, TO ENABLE E TO STROBE

Il verbo *to gate* ha di solito un solo significato:

To gate — Controllare il passaggio in un circuito digitale di un segnale digitale.

To enable e to strobe sono dei sinonimi:

To enable — Permette ad un circuito digitale di essere attivato mediante la sospensione di un segnale di inibizione.

To strobe Attivare un circuito digitale.

Il termine *to gate* è di solito associato al controllo di un treno di impulsi, specialmente nelle applicazioni strumentali.

GLI INGRESSI ENABLE E STROBE DEI CIRCUITI INTEGRATI

I circuiti integrati digitali possono essere suddivisi in alcune categorie:

- Piccola scala di integrazione (SSI)

  Chip che contengono fino a trenta o quaranta transistor. I chip con semplici porte, buffer e flip-flop sono in questa categoria.

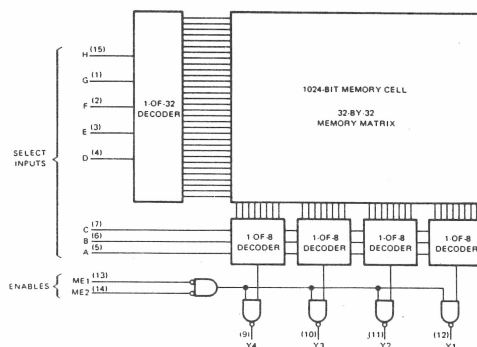
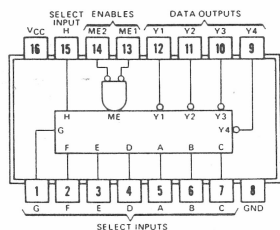
- Media scala di integrazione (MSI)

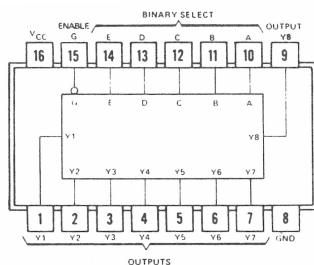
  Chip che si presentano come dei semplici sistemi logici, compresi contatori, multiplexer, demultiplexer, decodificatori, piccole memorie a lettura/scrittura, unità aritmetiche e registri di shift.

- Larga scala di integrazione (LSI)

  Circuiti integrati con una complessità tipica di centinaia o più porte o circuiti equivalenti. Il microprocessor in un singolo chip, che contiene fino a 5000 transistor, è in questa categoria, come il ricevitore/trasmettitore asincrono universale (UART); sono comprese anche le memorie a lettura/scrittura di un Kilobit, le ROM e le EPROM.

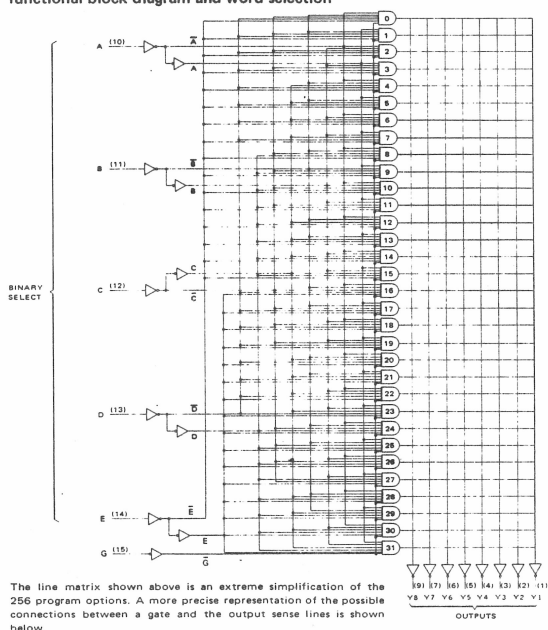
Se osservate i chip, vi accorgerete che molti hanno dei pin di ingresso di strobe ed enable. Faremo alcuni esempi:

Il chip 74187, memoria a sola lettura di 1024 bit ha un paio di ingressi enable che abilitano le porte di uscita della matrice di memoria,

Come imparerete in un successivo capitolo, le uscite dei chip di memoria vengono di solito collegati insieme *(BUSSED)*. La possibilità di selezionare, o abilitare (enable) un chip di memoria piuttosto che un altro è una delle caratteristiche più importanti delle grandi unità di memoria a semiconduttori.

Il chip 7488A, memoria a sola lettura di 256 bit, contiene un solo pin di ingresso enable che è detto "G".

La funzione base dell'ingresso enable al pin 15 risulta più chiara se vista in un diagramma a blocchi funzionale. L'ingresso G è collegato ad ognuna delle trentadue porte che decodificano le celle della matrice di memorie:

TYPES SN5488A, SN7488A
256-BIT READ-ONLY MEMORIES

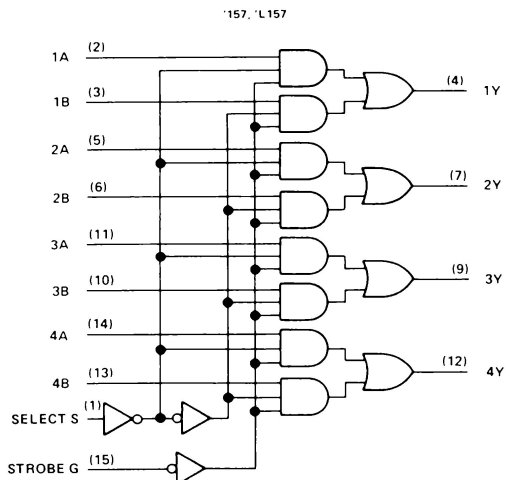
functional block diagram and word selection

The line matrix shown above is an extreme simplification of the 256 program options. A more precise representation of the possible connections between a gate and the output sense lines is shown below.

WORD-SELECT TABLE

WORD	INPUTS				
	E	D	C	B	A
0	L	L	L	L	L
1	L	L	L	L	H
2	L	L	L	H	L
3	L	L	L	H	H
4	L	L	H	L	L
5	L	L	H	L	H
6	L	L	H	H	L
7	L	L	H	H	H
8	L	H	L	L	L
9	L	H	L	L	H
10	L	H	L	H	L
11	L	H	L	H	H
12	L	H	H	L	L
13	L	H	H	L	H
14	L	H	H	H	L
15	L	H	H	H	H
16	H	L	L	L	L
17	H	L	L	L	H
18	H	L	L	H	L
19	H	L	L	H	H
20	H	L	H	L	L
21	H	L	H	L	H
22	H	L	H	H	L
23	H	L	H	H	H
24	H	H	L	L	L
25	H	H	L	L	H
26	H	H	L	H	L
27	H	H	L	H	H
28	H	H	H	L	L
29	H	H	H	L	H
30	H	H	H	H	L
31	H	H	H	H	H

H = high level, L = low level

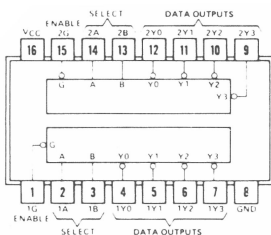
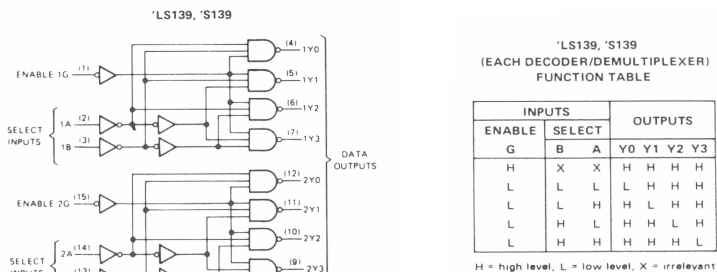
Il chip 74157, quadruple 2-line-to-1-line data selector, ha un solo pin di ingresso di strobe, STROBE G, che è collegato all'interno del chip alle otto porte AND a 3-ingressi.

TYPES SN54157, SN54L157, SN74157, SN74L157
QUADRUPLE 2-LINE-TO-1-LINE DATA SELECTORS/MULTIPLEXERS

Uno stato logico 0 a questo pin di ingresso disabilita il chip 74157.

Il chip 74139, decodificatore/demultiplexer, contiene due decodificatori/demultiplexer da 2 a 4 linee completamente indipendenti, ognuno dei quali ha il suo pin di ingresso enable, ENABLE 1G (pin 1) e ENABLE 2G (pin 15). Uno stato logico 0 al pin di ingresso enable abilita tutte le quattro porte NAND a 3-ingressi del decodificatore.

'LS139, 'S139
(EACH DECODER/DEMULTIPLEXER)
FUNCTION TABLE

INPUTS			OUTPUTS			
ENABLE	SELECT					
G	B	A	Y0	Y1	Y2	Y3
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

H = high level, L = low level, X = irrelevant

Gli esempi che abbiamo fatto dovrebbero essere sufficienti per dimostrare come un ingresso ad una porta, o ad un gruppo di porte, possa essere usato per attivare un circuito digitale. Quasi sempre uno stato logico 0 all'ingresso enable/strobe abilita il circuito integrato.

Supponendo che sia necessario fornire uno stato logico 0 all'ingresso enable/strobe di un circuito integrato, potreste chiedervi per quanto tempo bisogna lasciare questo stato logico 0 per abilitare il circuito integrato contenuto nel chip. La tabella 8 indica che è necessario un tempo compreso tra i 14 e i 22 ns.

Tabella 8. Tempo totale di ritardo di propagazione per gli ingressi enable/strobe per alcuni circuiti MSI e LSI.

Chip	Tempo totale di ritardo di propagazione, ns
7488 ROM	22
74LS139 decodificatore	21
74150 multiplexer	21
74154 decodificatore	19
74L154 decodificatore	38
74155 decodificatore	18
74157 multiplexer	14
74187 ROM	20

Il tempo necessario perchè uno stato logico 0 abiliti un certo circuito integrato dipende dalla complessità dei chip (quante porte ci sono tra l'ingresso e l'uscita) e dal fatto che il circuito usi una tecnologia bassa potenza (L), Schottky (S) o Schottky a bassa potenza (LS). Come si può vedere dalla tabella 7 il tempo è molto breve, circa 20 ns.

Abbiamo parlato degli ingressi enable/strobe perchè sono molto importanti quando si interfaccia un microcomputer. Nei capitoli seguenti imparerete che il microcomputer invia al suo esterno degli impulsi, della durata compresa tra 500 e 1333 ns, i quali forniscono uno strobe per il funzionamento dei circuiti integrati che si trovano ad interfacciare con esso. Benchè il microcomputer esegua queste operazioni ad una velocità troppo alta perchè possiate vedere qualcosa, gli impulsi da 500 a 1333 ns sono più che sufficienti se vengono confrontati con i tempi totali di ritardo di propagazione di 20 ns caratteristici dei chip MSI e LSI.

I VERBI: TO DISABLE (DISABILITARE) E TO INHIBIT (INIBIRE)

I verbi *disabilitare* o *inibire* esprimono il concetto opposto ad abilitare e fornire uno strobe (disable e enable - inhibit e strobe).

Disabilitare — Impedire il passaggio di un segnale digitale applicando un opportuno segnale al terminale disable di un dispositivo digitale.

Inibire — Impedire che avvenga una azione o che un dato digitale venga accettato o trasmesso. Si realizza mediante l'applicazione di un opportuno segnale all'apposito ingresso di un dispositivo digitale.

Quando incontrerete delle descrizioni del funzionamento dei circuiti integrati, troverete spesso frasi del tipo "abilitare il chip (Ingl. = chip enable)", "disabilitare il chip (Ingl. = chip disable)", "inibire il clock (Ingl. = clock inhibit)".

GLI AGGETTIVI: GATE, GATED GATING

Nei testi di elettronica digitale incontrerete frequentemente i termini *gate, gated* e *gating*. Questi termini verrano usati nel seguente modo:

gate circuit
gate pulse
gate signal

gated buffer
gated driver
gated counter

gating circuit
gating pulse
gating signal

Diamo adesso la definizione di ognuno di questi termini:

gate circuit — Circuito che lascia passare il segnale solo quando un impulso di gating viene applicato all'ingresso. Circuito elettronico con uno o più ingressi e un'uscita con la particolarità che un impulso va in uscita se, e solo se, è presente agli ingressi una determinata combinazione di impulsi.

gate pulse — Impulso che abilita un gate circuit affinchè lasci passare il segnale. Il gate pulse ha di solito una durata superiore a quella del segnale per essere sicuro che i tempi coincidano.

gate signal — Vedere gate pulse. Segnale che abilita un gate circuit affinchè lasci passare un segnale.

gated buffer — Circuito driver a bassa-impedenza che può essere usato come pilota nei multivibratori.

buffer — Elemento di circuito digitale che può essere usato per manipolare un fan-out elevato o per invertire i livelli di ingresso e uscita.

fan-out	Numero di carichi paralleli all'interno di una data famiglia logica, come la TTL, che possono essere guidati da una sola uscita di un circuito logico. Un chip TTL standard ha un fan-out uguale a 10, il che significa che può guidare dieci carichi TTL standard.
gated driver	Di solito è un driver che è sottomesso a una porta.
driver	Elemento di un circuito digitale accoppiato all'uscita di un circuito per aumentarne la capacità di guidare corrente o tensione, detta fan-out, del circuito. Per esempio, un clock driver viene usato per fornire la corrente necessaria per la linea di clock.
gating circuit	Circuito che opera come switch selettivo e permette il passaggio dei segnali solo durante alcuni intervalli di tempo in cui l'ampiezza del segnale è all'interno dei limiti prescritti.
gating pulse	Impulso che modifica o controlla il funzionamento di un gate circuit.
gating signal	Segnale digitale che modifica o controlla il funzionamento di un gate circuit.
trigger	Impulso che dà il via ad una azione.

SWITCH E PORTA: QUALE È LA DIFFERENZA?

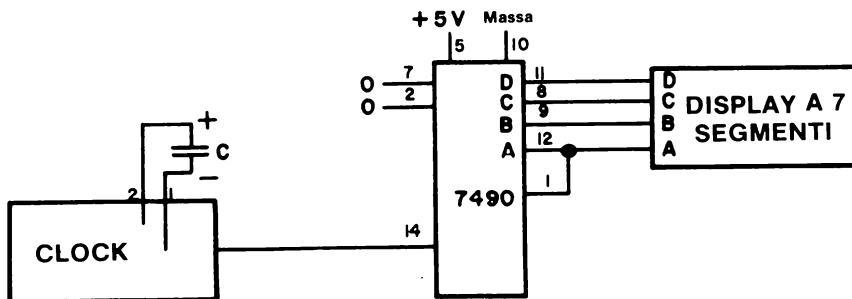
Diamo una definizione di switch:

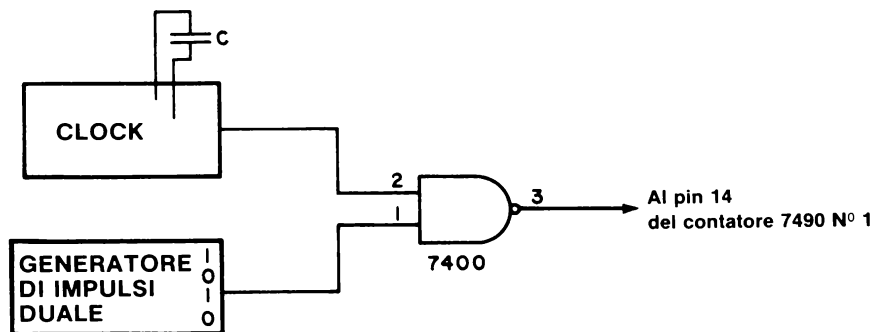
Switch	Dispositivo elettronico o meccanico che congiunge o interrompe una linea elettrica o che la commuta tra diverse possibili vie.

Malmstadt e Enke, nel loro libro "Digital Electronics", parlano delle seguenti differenze tra switch e gate:

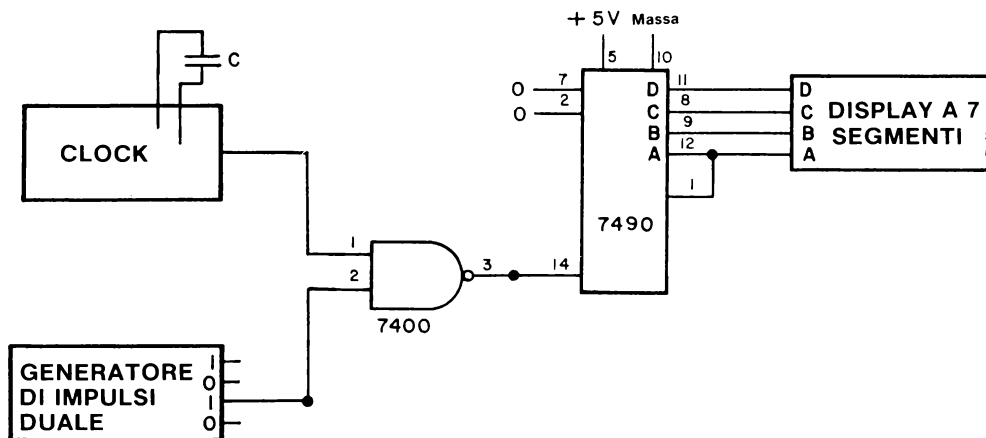
- Uno switch aperto impedisce il passaggio di un segnale, mentre una porta aperta permette il passaggio di un segnale.
- Una porta è un dispositivo a senso unico, mentre uno switch può far passare la corrente nei due sensi.
- Quando la porta è chiusa, il segnale non passa.
- Quando uno switch è chiuso il segnale può passare in ogni direzione.
- Quando una porta è aperta il segnale può passare dall'ingresso all'uscita, ma non in senso inverso.
- Quando uno switch è aperto, il segnale non può passare in nessuna direzione.
- Usando una porta, il passaggio dell'informazione digitale, nella forma di 0 logico o di 1 logico, avviene come livello di tensione di +5V o di massa.
- Usando uno switch il passaggio di corrente, di qualsiasi valore, avviene in tutti e due i sensi.

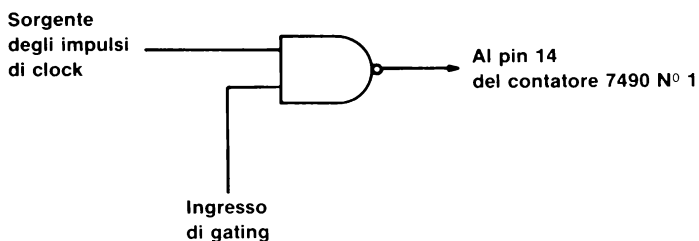
IL GATING DI UN CONTATORE

Nel capitolo 13, avete fatto un esperimento in cui avete misurato la frequenza dell'Outboard di clock in funzione della capacità che regola la scansione, e nell'Esperimento N. 2 avete usato un contatore a decade 7490,

ma avete avuto delle difficoltà quando la frequenza superava i 5 Hz. Mediante l'uso di alcuni contatori a decadi addizionali, è possibile misurare delle frequenze maggiori di 5000 Hz, come avete fatto nell'Esperimento N. 7. In ogni caso però, se è possibile fermare il contatore per vedere fino a che punto è arrivato, il problema non sussiste più. In questo paragrafo dimostreremo come si possa aggiungere questa funzione al circuito che effettua il conteggio.

Il problema di arrestare il conteggio è riconducibile al problema di trasformare il semplice contatore in un gated counter, in cui il gating di ingresso di una porta a 2-ingressi controlla se il segnale di clock giunge o no all'ingresso del contatore. Per esempio,

gli impulsi di clock verranno contati fino a che il pin della porta NAND a 2-ingressi 7400 è all'1 logico. Se premete il generatore di impulsi, il pin 1 passa allo stato logico 0 e viene bloccato il passaggio dell'impulso di clock;

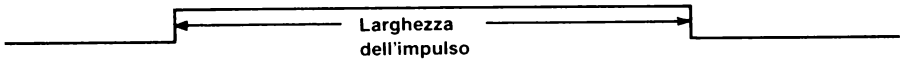
avete quindi disabilitato il contatore.

Qui sotto mostriamo, come esempio, un contatore a una decade gated. Al posto delle porte NAND a 2-ingressi si potrebbe usare una qualsiasi delle quattro porte base a 2-ingressi: AND, OR, NAND e NOR. Basta una porta indipendentemente da quante siano le decadi del contatore.

TIPI DI MISURE DEI CONTATORI

Il funzionamento di un gated counter può essere descritto in termini di ingresso e uscita associati con l'elemento di gating del contatore,

Usando la porta è possibile eseguire due tipi di misurazioni sperimentali.

- Data una certa sorgente di impulsi di frequenza conosciuta, è possibile misurare la *grandezza dell'impulso*, cioè la durata dell'impulso di gating.

- Dato un impulso di gating di una certa durata, è possibile misurare il numero di impulsi di clock che arrivano in un certo tempo.

Questi due tipi di misurazione possono essere compresi più facilmente con l'ausilio della forma d'onda digitale che mostriamo qui sotto.

Supponiamo che il contatore sia negative-edge triggered, allora si avranno sei conteggi prima che l'ingresso di gating ritorni allo stato logico 0. Se la sorgente degli impulsi di clock avesse una frequenza molto più alta, si avrebbero molti più conteggi con lostesso impulso di gating. Modificando la durata dell'impulso di gating e la frequenza della sorgente di impulsi di clock, si possono fare molte misurazioni. La sorgente degli impulsi di clock potrebbe essere,

- Un clock che opera ad una frequenza compresa tra 0,01 Hz e 10 MHz.
- Un generatore di impulsi.
- Uno switch meccanico non dotato di logica antirimbalzo.
- Un trasduttore tensione-frequenza che converte una quantità fisica in una serie di impulsi di clock, la cui frequenza è in qualche modo in relazione con la grandezza di questa quantità fisica. Usando una tecnica di questo tipo, è possibile misurare, frequenze, capacità, resistenze, tensioni, corrente, velocità, eventi, etc.

La sorgente dell'impulso di gating potrebbe essere,

- Un clock funzionante ad una frequenza molto più bassa, da 0,01 Hz a 10 Hz.
- Un multivibratore monostabile.
- Un generatore di impulsi dotato di logica antirimbalzo.
- Altri sistemi digitali che generano un singolo impulso di clock.

La sorgente più importante è il multivibratore monostabile, di cui parleremo nel prossimo capitolo.

INTRODUZIONE AGLI ESPERIMENTI

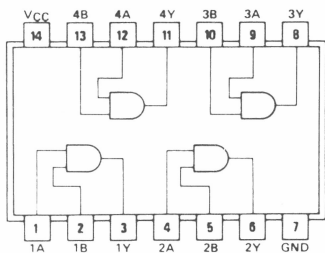
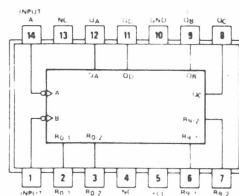
I seguenti esperimenti mostrano i principi di funzionamento di un contatore gated.

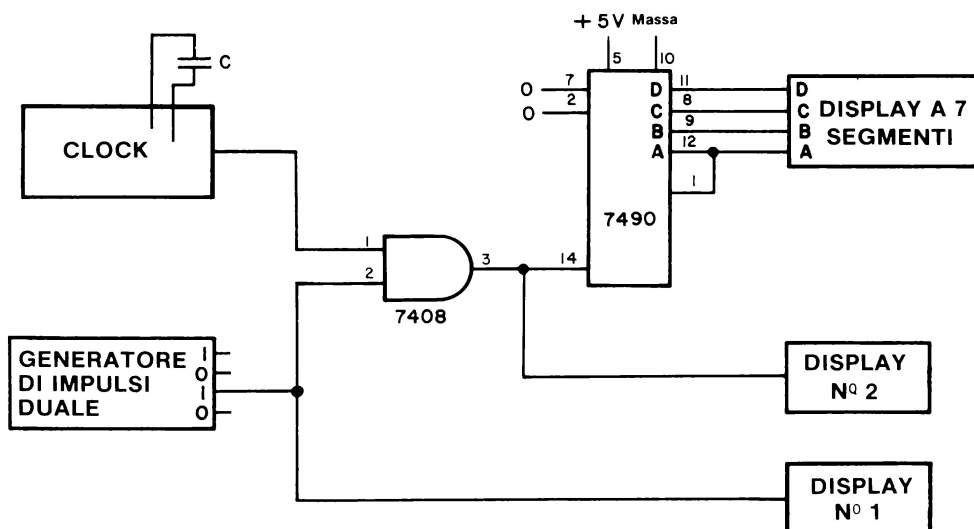
Esperimento N.	Commento
1	Mostrare come i chip 7400, 7408 e 7432 possano essere usati come gate per i chip 7490 decade counter e 7493 binary counter.
2	Mostrare come si usa un contatore a più decadi "gated" per misurare la frequenza dell'Outboard di clock in funzione della capacità che regola la scansione.

ESPERIMENTO N. 1

Scopo

Lo scopo di questo esperimento è di mostrare le operazioni di porte base come elementi di gating per un singolo contatore a decade 7490.

Configurazione dei pin del circuito integrato

7408

7490

Schema del circuito

Passo 1

Nello schema, il display potrebbe essere un singolo indicatore a LED o un display zero/uno fatto con un Outboard display a sette segmenti.

Collegate il circuito. Così come è fatto, la porta AND è abilitata o disabilitata?

È abilitata. L'ingresso di gating viene dal generatore di impulsi. Siccome viene usata l'uscita del generatore di impulsi all'1 logico, il pin 2 del chip 7408 è all'1 logico fino a che non si preme il generatore di impulsi.

Passo 2

Alimentate il circuito e usate una frequenza di clock lenta, di circa 1 Hz. Cosa vedete sul display a sette segmenti e sul display N. 2?

Vediamo sul display a sette segmenti una sequenza di conteggi e sul display N.2 vediamo l'indicatore a LED che lampeggia.

Passo 3

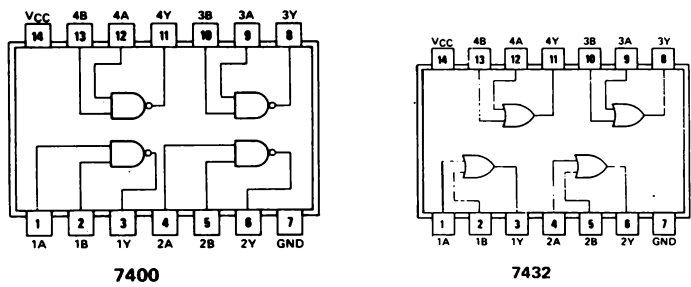
Premete il generatore di impulsi. Cosa succede?

Il contatore a decade arresta il conteggio, il display N. 2 resta allo stato logico 1 e il display N. 1 resta allo 0 logico. Abbiamo disabilitato la porta.

Passo 4

Eseguite lo stesso esperimento con una porta NAND 7400 e con una porta OR 7432 e mettete i risultati nella tabella seguente:

Porta	Stato logico dell'ingresso di gating	Porta abilitata o disabilitata	Conta o non conta?	Stato logico del display N. 1	Stato logico del display N. 2
7408	0				
	1	abilitato	conta	1	impulsi di clock
7400	0				
	1				
7432	0				
	1				

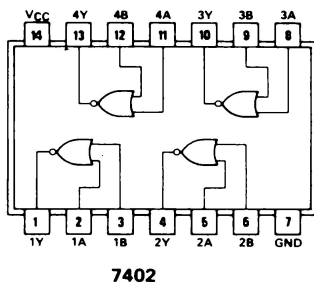
La configurazione dei pin dei circuiti integrati 7400 e 7432 sono le seguenti:

7400

7432

Dovreste aver notato che sono compatibili col chip 7408 pin per pin, quindi per sostituire un chip con l'altro, basta togliere l'alimentazione e sostituire il chip 7408 con il 7400 o il 7432.

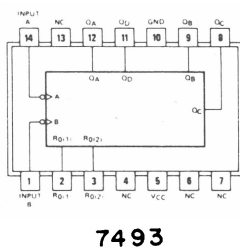
Quando abbiamo fatto questo esperimento abbiamo ottenuto i seguenti risultati:

Porta	Stato logico dell'ingresso di gating	Porta abilitata o disabilitata?	Conta o non conta?	Stato logico del display N. 1	Stato logico del display N. 2
7408	0	disabilitato	no	0	0
	1	abilitato	si	1	impulsi di clock
7400	0	disabilitato	no	0	1
	1	abilitato	si	1	impulsi di clock
7432	0	abilitato	si	0	impulsi di clock
	1	disabilitato	no	1	1
7402	0				
	1				

Completate la tabella con quello che prevedete debba essere il comportamento della porta NOR 7402. Guardando la configurazione dei pin data di seguito, quali cambiamenti dovete apportare al circuito illustrato nello schema?

7402

Il chip 7402 non è compatibile, pin per pin, con i chip 7400, 7408 e 7432. I due ingressi si trovano adesso al pin 2 e 3 e l'uscita si trova al pin 1.

Passo 5

Il chip 7493, contatore binario,

7493

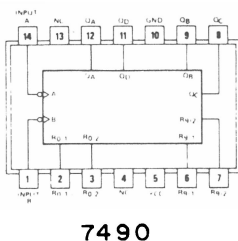
è compatibile pin per pin col chip 7490, contatore a decade, sostituite quindi il 7490 col 7493 e ripetete i Passi 2 e 3 con una porta di vostra scelta. In sostanza, il contatore binario si comporta come quello a decade?

Si. Si possono osservare sedici stati unici di conteggio invece di dieci col dsplay a sette segmenti si vedono cinque insoliti simboli e un blank oltre ai numeri tra 0 e 9.

ESPERIMENTO N. 2

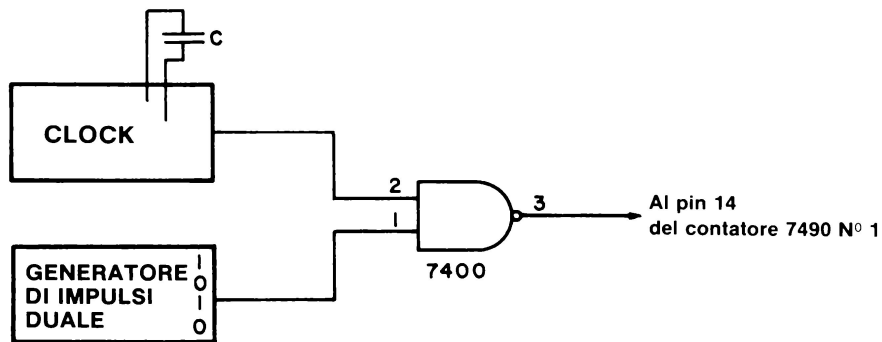
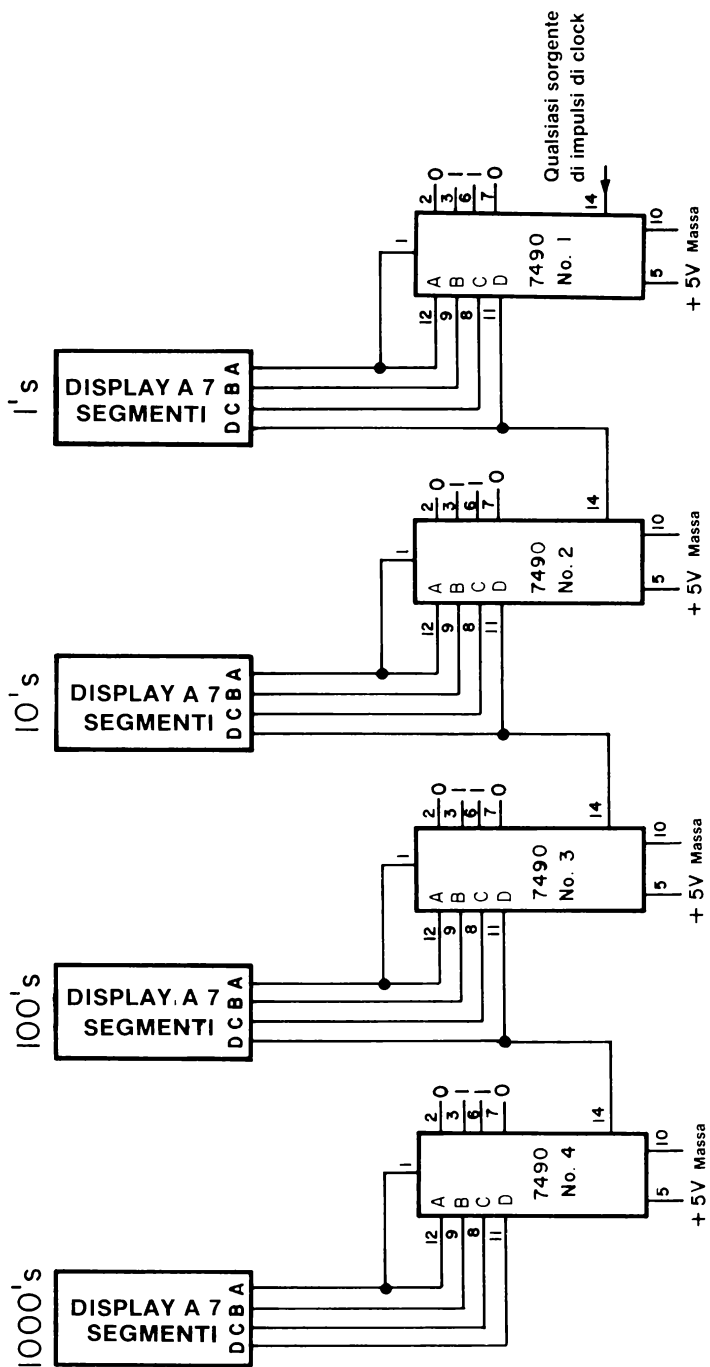
Scopo

Lo scopo di questo esperimento è di usare un *contatore a più decadi gated* per misurare la frequenza dell'Outboard di clock in funzione della capacità di scansione. Questo esperimento è essenzialmente identico all'Esperimento N. 7 del capitolo 13.

Configurazione dei pin del circuito integrato

7490

Schema del circuito

Lo schema del contatore a quattro decadi è riportato successivamente. Dovete fare in modo di azzerare simultaneamente tutti i contatori a decade 7490, come abbiamo mostrato nell'Esperimento N. 7 del capitolo 13. Il circuito di gating base è il seguente:

1000's
100's
10's
1's
DISPLAY A 7
SEGMENTI
D C B A
7490
No. 4
7490
No. 3
7490
No. 2
7490
No. 1
Qualsiasi sorgente
di impulsi di clock
+ 5V
Massa

Passo 1

Realizzate il circuito. Più decadi avete e più si allarga la banda di frequenza che potete misurare.

Passo 2

Fate variare la capacità di scansione dell'Outboard di clock e completate la seguente tabella con i risultati che ottenete.

Capacità di scansione in μF	Conteggi/ minuto	Frequenza conteggi/s	Frequenza^{-1} s/conteggio	Frequenza x Capacità 10^{-8} Farad/s
Se avete solo tre decadi nel contatore, le seguenti frequenze potrebbero essere troppo alte per essere misurate.				
nessuno				
0,00001				
0,00002				
0,00005				
0,0001				
0,0002				
Le seguenti, invece, dovreste riuscire a misurarle.				
0,0005				
0,001				
0,002				
0,005				
0,01				
0,02				
0,05				
0,10				
0,22				
0,47				
1				
2,2				

Basandovi sui risultati ottenuti, calcolate la capacità interna del clock che state usando. Non potete fare questo calcolo se non riuscite a misurare la frequenza di clock senza la capacità di scansione.

DOMANDE RIEPILOGATIVE

Le seguenti domande servono ad aiutarvi a ripassare i concetti di gating e di uso delle porte per applicazioni di tipo gating.

1. Quale stato logico è necessario fornire all'ingresso di gating per abilitare le seguenti porte?

   a. porta NOR a 2-ingressi

   b. chip 7408

   c. porta OR-Esclusivo

   d. chip 7400

   e. porta NAND a 2-ingressi

   f. porta OR a 2-ingressi

   g. chip 7432

2. Quanti ingressi di gating ci sono sui seguenti chip? Fate riferimento al capitolo 10 per avere la descrizione di questi chip.

   a. chip 7430

   b. chip 7411

   c. chip 7420

   d. chip 7410

   e. chip 7427

   f. chip 7404

   g. chip 7486

   h. chip 7451

   i. chip 7432

3. Nello spazio qui sotto disegnate i collegamenti addizionali richiesti per convertire un contatore a quattro decadi illustrato nell'Esperimento N. 2 in un contatore a 5 decadi.

4. Qual'è lo stato di riposo dei seguenti circuiti quando sono disabilitati?

   a. chip 7410

   b. porta NOR a 3-ingressi

   c. porta NAND a 2-ingressi

   d. chip 7430

   e. chip 7486

   f. chip 7408

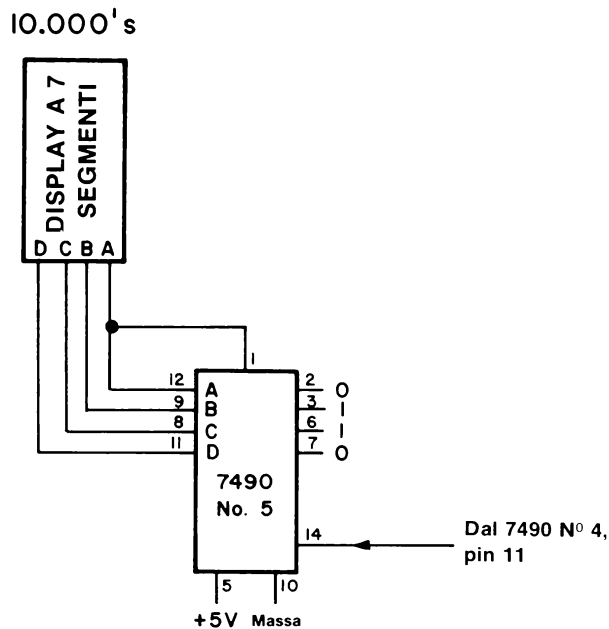
   g. chip 7402

   h. porta OR a 2-ingressi

RISPOSTE

1. a. 0 logico
   b. 1 logico
   c. non applicabile
   d. 1 logico
   e. 1 logico
   f. 0 logico
   g. 0 logico

2. a. sette
   b. due
   c. tre
   d. due
   e. due
   f. non applicabile
   g. uno, ma il chip si comporta come un invertitore controllabile.
   h. non applicabile
   i. uno

3. Lo schema è il seguente:

4. a. 1 logico
   b. 0 logico
   c. 1 logico
   d. 1 logico
   e. Non esiste stato di riposo per la porta OR-Esclusivo
   f. 1 logico
   g. 0 logico
   h. 1 logico

CAPITOLO 15

MULTIVIBRATORI MONOSTABILI E ASTABILI

INTRODUZIONE

In questo capitolo vi introdurremo all'uso dei multivibratori monostabili per fornire degli impulsi di gating abbastanza precisi nei sistemi di contatori gated. Collegherete due monostabili, il 74121 - che può generare un impulso monostabile molto breve ma che non è molto stabile - e il 555 - che genera degli impulsi più lunghi e che invece è molto stabile. Collegherete anche un multivibratore astabile, cioè un clock, usando il chip 555.

OBIETTIVI

Alla fine di questo capitolo sarete in grado di:

- Dare la definizione di elemento astabile, elemento monostabile ed elemento bistabile.
- Descrivere le caratteristiche di funzionamento dei chip monostabili 74121, 74122 e 74123.
- Spiegare quali sono le differenze di funzionamento del chip 555 come multivibratore astabile e monostabile.
- Misurare la frequenza di un clock avendo un circuito multivibratore monostabile affidabile.
- Misurare la lunghezza di un impulso di un multivibratore monostabile avendo un clock di frequenza nota.

MULTIVIBRATORI MONOSTABILI

I circuiti integrati *multivibratori monostabili* vengono usati per generare degli ingressi di gating ai circuiti di contatori gated. Sono considerati dei chip *ibridi* analogico-digitali in cui l'uscita del circuito digitale viene determinata dalla costante di tempo RC di un circuito analogico che è collegato esternamente al chip.

Un multivibratore monostabile è un *elemento monostabile* che si differenzia dagli *elementi astabili* e *bistabili* per quanto detto nelle seguenti definizioni.

Elemento astabile	È un elemento a due stati che non ha uno stato stabile.
Elemento bistabile	Altro nome del flip-flop. È un circuito in cui l'uscita ha due stati stabili e può essere spostata da una all'altra da segnali in ingresso, ma che rimane indefinitamente in uno stato anche dopo che il segnale in ingresso viene tolto.
Elemento monostabile multivibratore monostabile	È un circuito che ha un solo stato stabile dal quale può essere spinto a cambiare stato, ma solo per un determinato periodo di tempo dopo il quale torna allo stato originario. È detto anche multivibratore on-shot o single-shot o start-stop.

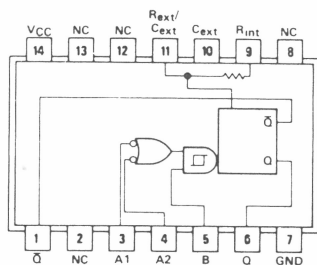
Gli elementi astabili possono passare da uno stato all'altro e vengono usati come clock o oscillatori. Gli elementi bistabili possono essere spinti da uno stato all'altro e possono rimanere permanentemente in questo stato anche dopo che il segnale è stato tolto. Vengono usati come latch e memorie. Gli elementi monostabili hanno un solo stato stabile; questi elementi possono essere spinti in un certo stato, ma ci rimangono solo per un periodo di tempo limitato dopo di che tornano allo stato iniziale o stabile. L'uso principale dei multivibratori monostabili è *la generazione di impulsi conosciuta da impulsi di durata più breve, più lunga o comunque sconosciuta.*

Conoscerete quattro circuiti integrati multivibratori monostabili:

- Timer 555 collegato come multivibratore monostabile.
- Multivibratore monostabile 74121 con ingresso trigger Schmitt.
- Multivibratore monostabile "retriggerable" con clear 74122.
- Doppio multivibratore monostabile "retriggerable" con clear 74123.

Il timer 555 è il migliore per impulsi che vanno da qualche microsecondo alle ore e per applicazioni in cui la lunghezza dell'impulso debba essere nota con una precisione superiore allo 0,05%. I monostabili 74121, 74122 e 74123 vengono preferiti per impulsi tra 40 ns e 10 μs. Se volete dal timer 555 una precisione dello 0,05%, ricordatevi che dovete usare dei componenti passivi che siano almeno altrettanto precisi; questi componenti hanno anche dei bassi coefficienti di temperatura. In pratica potete riuscire ad avere una precisione dello 0,3%.

MULTIVIBRATORE MONOSTABILE **74121**

Il multivibratore monostabile 74121 ha la seguente configurazione di pin e tabella della verità:

121

FUNCTION TABLE

INPUTS			OUTPUTS	
A1	A2	B	Q	$\bar{Q}$
L	X	H	L	H
X	L	H	L	H
X	X	L	L	H
H	H	X	L	H
H	.	H	⎍	⊔
.	H	H	⎍	⊔
.	↓	H	⎍	⊔
L	X	.	⎍	⊔
X	L	.	⎍	⊔

NOTES A. H = high level (steady state), L = low level (steady state), ↑ = transition from low to high level, . = transition from high to low level, ⎍ = one high level pulse, ⊔ = one low level pulse, X = irrelevant (any input, including transitions).

B. To use the internal timing resistor of '121, 'L121, '122, or 'L122, connect R_{int} to V_{CC}.

C. An external timing capacitor may be connected between C_{ext} and R_{ext} C_{ext} (positive).

D. For accurate repeatable pulse widths, connect an external resistor between R_{ext} C_{ext} and V_{CC} with R_{int} open circuited.

E. To obtain variable pulse widths, connect external variable resistance between R_{int} or R_{ext} C_{ext} and V_{CC}.

Le note vanno bene anche per i chip 74121, 74122 e 74123 che verranno descritti in seguito. NC significa nessuna connessione. Il TTL Data Book della Texas Instruments Incorporated dà una ottima descrizione del chip 74121 che riportiamo qui sotto. Una delle caratteristiche più importanti del chip 74121 è che *l'impulso d'ingresso può essere di qualsiasi durata rispetto all'impulso di uscita*. La ripetitività degli impulsi di uscita è di circa lo 0,5%. Lo Schmitt-trigger input al pin 5 permette di usare nel chip delle forme d'onda digitali che cambiano lentamente. Per altri dettagli consultate il capitolo 10 del Bugbook II. Il 74121 non è "retriggerable"; una volta che è cominciato il suo periodo di tempo RC, continua senza essere influenzato da altri ingressi di trigger.

Descrizione

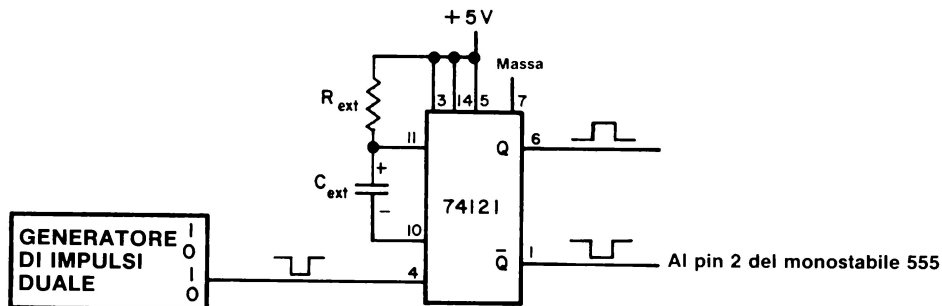
Questi multivibratori hanno due ingressi negative-transition-triggered e un solo ingresso positive-transition-triggered che può essere usato come ingresso di inibizione. Vengono forniti impulsi di uscita complementari.

L'azione di trigger avviene ad un particolare livello di tensione e non è in relazione al tempo di passaggio di stato dell'impulso di ingresso. La circuiteria dello Schmitt-trigger input (isteresi TTL) dell'ingresso B fornisce un trigger jitter-free dagli ingressi come velocità di passaggio di stato più lenta di 1 V/s, dando un circuito con una buona immunità al rumore, di solito di 1,2 V. C'è anche una buona immunità con riferimento al Vcc, di solito di 1,5 V, grazie alla circuiteria di latch interna.

Una volta accese le uscite sono indipendenti da eventuali cambiamenti degli ingressi e sono funzione solo dei componenti di timing. Gli impulsi di ingresso possono essere di una durata qualsiasi rispetto agli impulsi di uscita. La lunghezza degli impulsi di uscita può essere variata da 40 ns a 28 s scegliendo gli opportuni componenti di timing. Senza componenti di timing esterni (cioè Rint collegata a Vcc, Cext e Rext/Cext aperti) viene fornito un impulso di uscita di 30 o 35 ns che può essere usato come segnale di reset d-c triggered. I tempi di salita e discesa dell'uscita sono compatibili TTL e indipendenti dalla durata dell'impulso.

La durata dell'impulso viene fornita per mezzo di una compensazione interna ed è virtualmente indipendente dalla Vcc e dalla temperatura. In molte applicazioni la stabilità dell'impulso è limitata solo dall'accuratezza dei componenti temporizzatori esterni.

La funzione jitter-free è mantenuta in tutto il range di temperatura e Vcc e compresa tra sei decadi di capacità di timing (da 10 pF e 10 μF) e oltre una decade di resistenze di timing (da 2kΩ a 30kΩ per il SN54121/SN54L121 ed da 2 kΩ a 40 kΩ per il SN74121/SN74L121). All'interno di questi valori, la durata dell'impulso viene determinata dalla seguente relazione: Tw (out) = Ct Rt ln 2≈ 0,7 Ct Rt. Nei circuiti in cui non sia critico il cut-off, bisogna usare capacità di più di 1000μF e resistenze di scansione minore di 1,4 kΩ. Inoltre la possibilità di variazione della durata dell'impulso di uscita jitter-free viene ampliata se la Vcc è di 5 V e la temperatura esterna è di 25° C. I Duty cycle maggiori del 90% vengono raggiunti quando si usano i massimi valori ammessi per Rt.

Uno dei modi di collegare sul breadboard il chip 74121 è il seguente:

Osservate il commento posto a fianco dell'uscita Q, "al pin 2 del monostabile 555". Il monostabile 555, di cui parleremo tra poco, si differenzia dal chip 74121 per il fatto che *l'impulso di ingresso deve essere di durata minore dell'impulso di uscita.* Quindi se volete sfruttare i vantaggi delle caratteristiche del monostabile 555, dovete pilotare il suo ingresso con un chip 74121.

I valori di R_{ext} e C_{ext} dello schema si possono ricavare dai seguenti grafici:

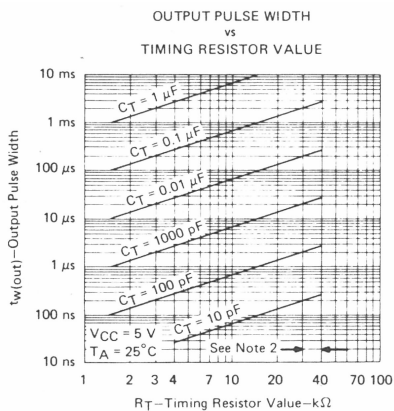
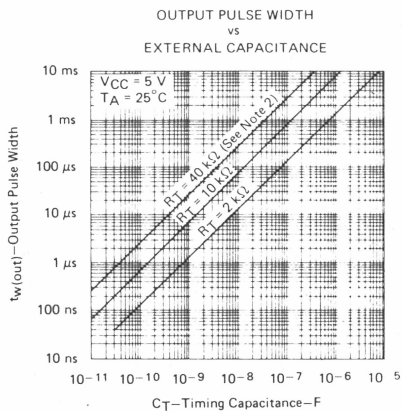



FIGURE 6

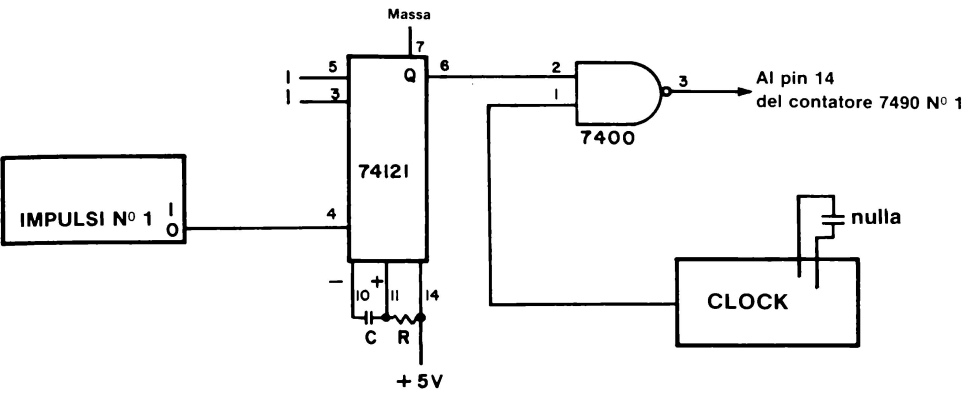
FIGURE 7

§ Data for temperatures below 0°C and above 70°C are applicable for SN54121 and SN54L121 only.

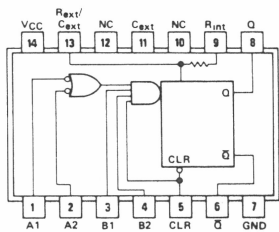
NOTE 2: These values of resistance exceed the maximum recommended for use over the full temperature range of the SN54121 and SN54L121.

Notate che il valore massimo di R_{ext} è di soli 40.000 Ω. La durata dell'impulso può essere calcolata con la seguente equazione:

$$t_{\omega} = 0,693 \ R_{ext} \ C_{ext}$$

dove R_{ext} è in ohm e C_{ext} è in Farad. La durata dell'impulso, t_{ω} è in secondi.

Quando il chip 74121 deve essere usato come sorgente di impulsi di gating per un contatore gated, va collegato nel seguente modo:

L'uscita dalla porta NAND a 2-ingressi 7400 deve essere collegata all'ingresso di una cascata di contatori a decade 7490, come abbiamo mostrato nel capitolo 13.

MULTIVIBRATORE MONOSTABILE RETRIGGERABLE 74122

La configurazione dei pin e la tabella della verità del multivibratore monostabile retriggerable con clear 74122 sono le seguenti:

122

FUNCTION TABLE

INPUTS					OUTPUTS	
CLEAR	A1	A2	B1	B2	Q	Q̄
L	X	X	X	X	L	H
X	H	H	X	X	L	H
X	X	X	L	X	L	H
X	X	X	X	L	L	H
X	L	X	H	H	L	H
H	L	X	↑	H	⊓	⊔
H	L	X	H	↑	⊓	⊔
H	X	L	H	H	L	H
H	X	L	↑	H	⊓	⊔
H	X	L	H	↑	⊓	⊔
H	H	↓	H	H	⊓	⊔
H	↓	↓	H	H	⊓	⊔
H	↓	H	H	H	⊓	⊔
↑	L	X	H	H	⊓	⊔
↑	X	L	H	H	⊓	⊔

Il termine "retriggerable" si riferisce alla possibilità di ridare un ingresso di trigger prima ancora che l'impulso di uscita sia finito, e l'effetto che si ottiene è di estendere di un periodo la durata del segnale di uscita; inoltre il processo è ripetibile senza limitazione. Questa possibilità permette di generare impulsi di una durata molto lunga. Tutti i particolari dell'impulso di retrigger sono descritti nella pagina del TTL Data Book della Texas Instruments Incorporated che riportiamo qui sotto.

I monostabili "retriggerable" vengono usati per scoprire eventuali malfunzionamenti dei sistemi digitali sincroni o clocked che lavorano con una sorgente di impulsi di clock centralizzata. Rientrano in questa categoria i computer e gli strumenti digitali. La costante di tempo RC dei monostabili retriggerable può essere modificata in modo che l'uscita Q resti all'1 logico fino a che il clock continua a fornire impulsi all'ingresso di trigger. Se per una qualsiasi ragione si interrompono gli impulsi di clock, l'uscita passa allo 0 logico e dà il via ad una sequenza di eventi per staccare il computer o lo strumento digitale in modo da salvare il contenuto della memoria prima che venga a mancare completamente la corrente, o in modo di staccare l'alimentazione da una macchina che scalda troppo. Questo monostabile è il 74123 possono essere sottoposti a clear, cioè riportati al loro stato normale con un impulso di clear; inoltre è possibile, con l'ingresso clear, fornire un trigger al monostabile. Gli ultimi due stati della tabella della verità dimostrano questo fatto.

Descrizione

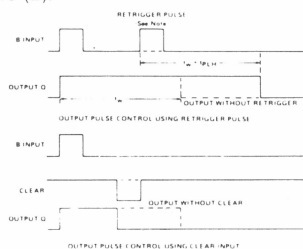
I multivibratori '122, '123, 'L122 e 'L123 hanno degli ingressi di trigger d-c gated low-level-active (A) e high-level-active (B), ed hanno anche overriding direct clear input. La possibilità retrigger permette di generare impulsi di durata anche molto lunga; infatti, fornendo un trigger prima che l'impulso di uscita sia terminato, questo viene esteso di un periodo. L'ingresso di overriding permette di interrompere un impulso in uscita in un qualsiasi momento indipendentemente dalle costanti di tempo R e C. La Figura A, che riportiamo qui sotto illustra il triggering con gli ingressi high-level-active (B).

NOTE: Retrigger pulse must not start before 0.22 C_{ext} (in picofarads) nanoseconds after previous trigger pulse.

FIGURE A–TYPICAL INPUT/OUTPUT PULSES

'122, '123
TYPICAL OUTPUT PULSE WIDTH
vs
EXTERNAL TIMING CAPACITANCE

FIGURE B

Questi monostabili sono fatti per fornire al progettista la massima flessibilità nel controllare la durata dell'impulso, allungandolo per mezzo del retriggering o accorciandolo col clear. Il '122 e il 'L122 hanno al loro interno una resistenza che permette, se lo si vuole, di far funzionare il circuito con un solo condensatore esterno. Per le applicazioni che richiedono un alta precisione della durata dell'impulso (fino a 28 s) e non hanno bisogno del clear, è meglio usare il chip '121 e 'L121.

L'impulso di uscita è funzione principalmente del condensatore e della resistenza esterna. Con Cext > 1000 pF, la durata dell'impulso di uscita $t\omega$ viene definita dalla seguente relazione:

$$t\omega = K \cdot Rt \cdot Cext \left(1 + \frac{0,7}{Rt}\right)$$

dove

Rt è in KΩ (resistenza interna o esterna),
Cext è in pF,
$t\omega$ è in ms,
K è 0,32 per il '122; 0,28 per il '123; 0,37 per il 'L122 e 0,33 per il 'L123.

Per durate d'impulso con Cext ≤ 1000 pF, fare riferimento alle Figure B e C.

'L122
TYPICAL OUTPUT PULSE WIDTH
vs
EXTERNAL TIMING CAPACITANCE

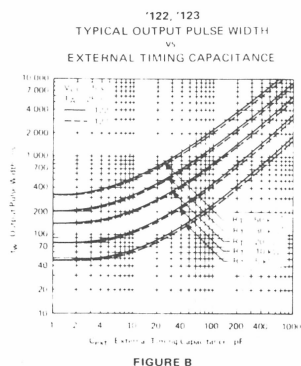
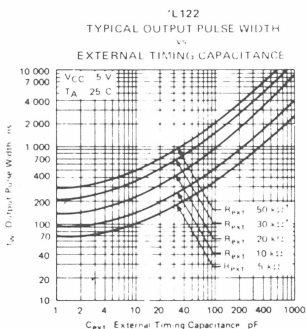
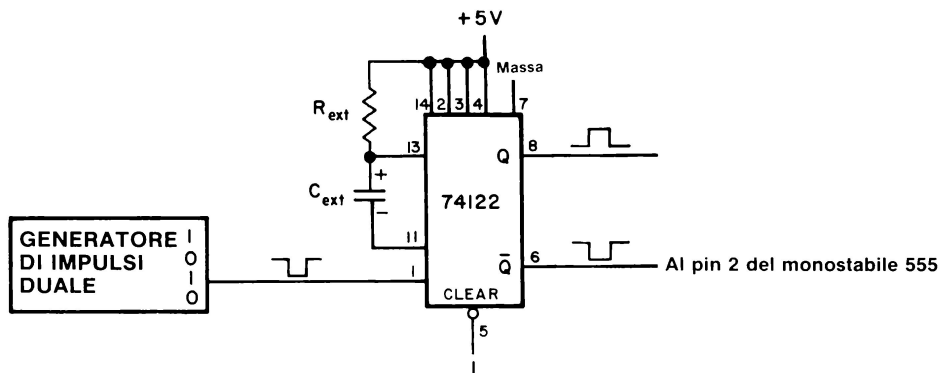
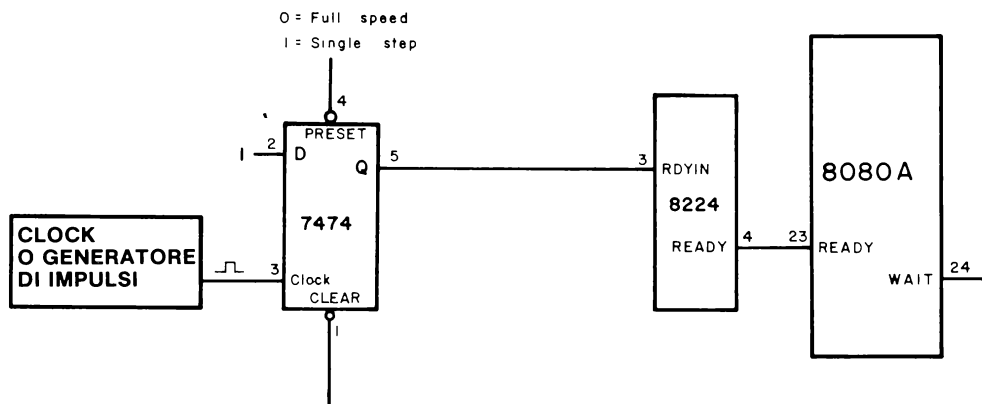
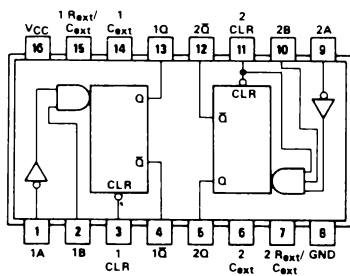



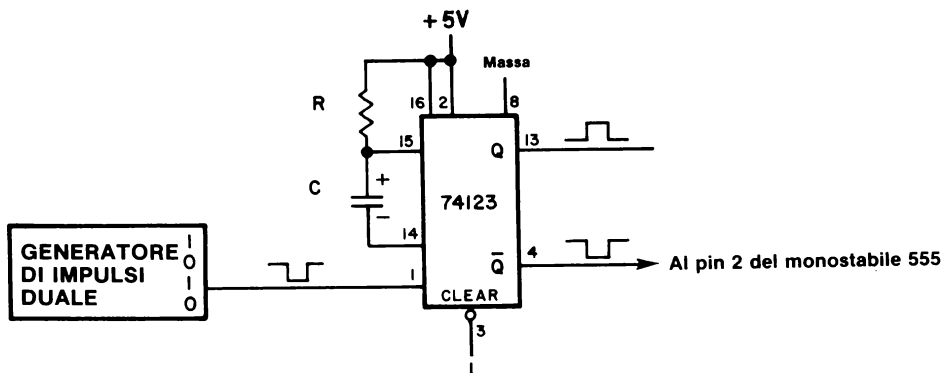
FIGURE C

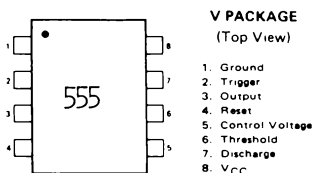
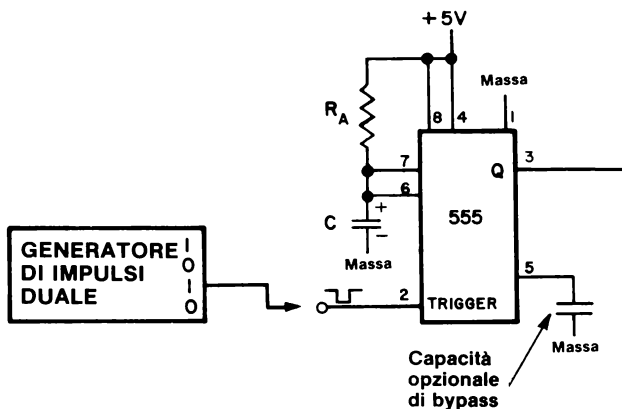
*These values of resistance exceed the maximum recommended for use over the full temperature range of the SN54 and SN54L circuits.

Bisogna prendere delle precauzioni speciali quando la capacità di timing è più grande di 1000 pF e si usa l'ingresso clear; allora bisogna inserire un diodo nel circuito, come mostriamo qui sotto. Il diodo cambia la formula che determina il tempo.

Un metodo per collegare il chip 74122 è mostrato di seguito:

Col metodo del 'sbagliando s'impara' abbiamo determinato gli esatti valori delle resistenze e della capacità di timing che permettono di costruire un circuito single-step per il microcomputer MMD-1; questo circuito usa un monostabile 74122. *Invece di questo circuito, vi proponiamo piuttosto il seguente, che non ha bisogno del chip 74122.*

DOPPIO MULTIVIBRATORE MONOSTABILE RETRIGGERABLE 74123

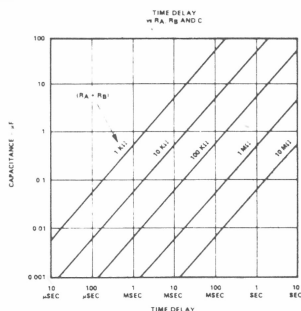
Il doppio multivibratore monostabile retriggerable 74123,

INPUTS			OUTPUTS	
CLEAR	A	B	Q	Q̄
L	X	X	L	H
X	H	X	L	H
X	X	L	L	H
H	L	↑	⎍	⊔
H	↓	H	⎍	⊔
↑	L	H	⎍	⊔

contiene due monostabili indipendenti che si comportano come il monostabile rettriggerable 74122. Il monostabile 74123 può essere collegato sul breadboard nel seguente modo:

MULTIVIBRATORE MONOSTABILE 555

Il circuito integrato timer 555 è un chip ibrido analogico-digitale che può funzionare sia come oscillatore astabile che come multivibratore monostabile. Il chip ha solo otto pin

e viene collegato come mostrato di seguito per produrre un monostabile che ha una buona stabilità e ripetibilità,

La relazione che dà la durata dell'impulso $t\omega$ in funzione delle costanti di tempo, è la seguente:

$$t\omega = 1{,}1\ R_A\ C$$

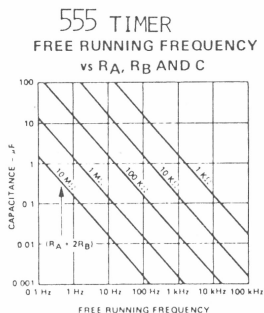
dove R_A è in ohm, C è in Farad e $t\omega$ è in secondi. La durata minima dell'impulso generato da monostabile 555 è di alcuni microsecondi. Una delle caratteristiche più importanti del chip è che *l'impulso in ingresso deve essere più breve dell'impulso di uscita.* Quindi se volete produrre un impulso monostabile di 10 ns, basta che troviate qualcosa che possa fornire un trigger al monostabile con un impulso anche più breve di 10 ns. La ripetitività della lunghezza dell'impulso del monostabile 555 è almeno dello 0,1%.

Il grafico che riportiamo, per concessione della Signetics Corporation, mostra la capacità richiesta per produrre impulsi di uscita varianti tra 10 ns e 10 s per mezzo di una serie di resistori di timing.

Raccomandiamo i seguenti valori di capacità e resistenze timing:

Capacità di timing	minima 500 pF; massimo limitato dalla dispersione del condensatore.
Resistenza di timing	minimo 1000 Ω; massimo 3,3 MΩ.

L'impulso di trigger massimo che raccomandiamo è di 1/4 dell'impulso di uscita. Di solito suggeriamo di usare i chip monostabili 74121, 74122 e 74123 col monostabile 555 perchè si possono produrre impulsi con la ripetitività più grande che con tutti gli altri monostabili della serie 7400.

MULTIVIBRATORE ASTABILE 555

Il chip timer 555 può essere collegato in modo da diventare un multivibratore astabile, cioè un clock; di seguito ne mostriamo il circuito. Il grafico fornito dalla Signetics Corporation dà la capacità richiesta per produrre la frequenza di uscita desiderata in funzione della somma $R_A + 2 R_B$:

La frequenza di clock può essere calcolata usando la seguente equazione,

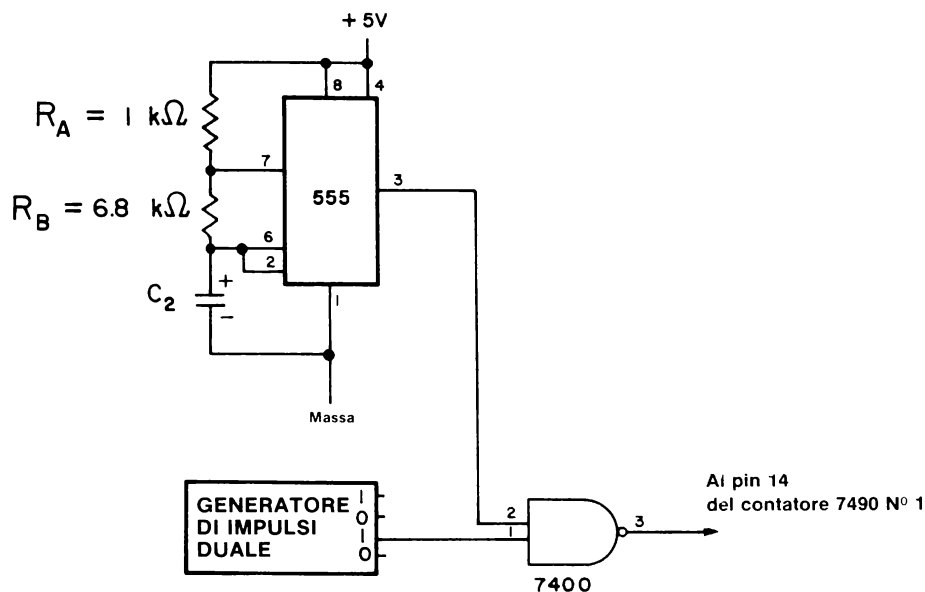
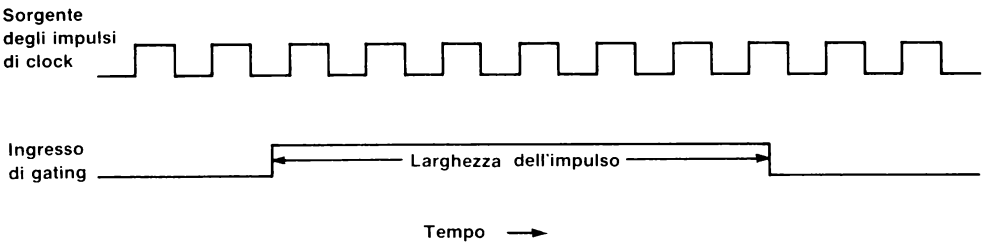
$$\nu = \frac{1,443}{(R_A + 2\,R_B)\,C}$$

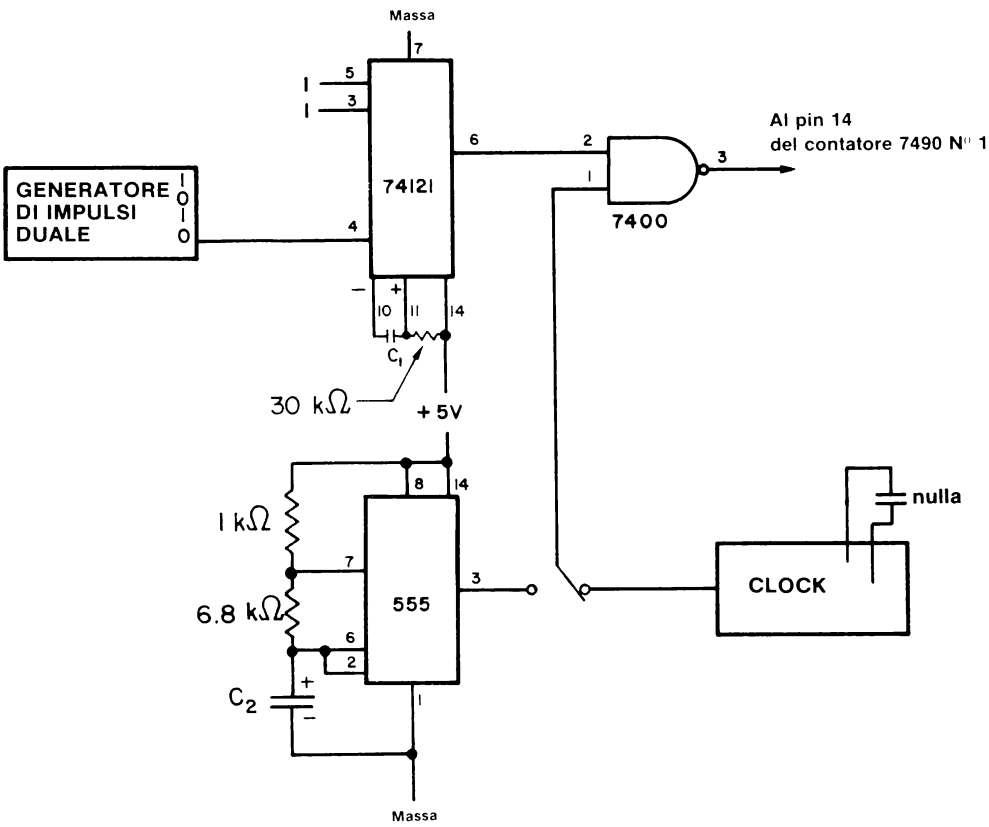
dove R_A e R_B sono in Ω, C è in Farad e ν è in Hz. Il *duty cicle* che in questo caso è il rapporto tra il tempo passato nello stato logico 0 e la somma dei tempi passati nello stato logico 0 e nello stato logico 1, è dato dalla seguente equazione:

$$D = \frac{R_B}{(R_A + 2\,R_B)}$$

Se si vuole avere in uscita un'onda quadra simmetrica, allora D=0,50; o $R_A \ll R_B$ Per l'outboard di clock, R_A = 100 kΩ e R_B = 1 MΩ. Un tipico circuito per il multivibratore astabile 555 è indicato di seguito in cui R_A = 1 kΩ e R_B = 6,8 kΩ. La frequenza di clock diventa,

$$\nu = \frac{1,443}{(14,6\Omega)\;(1,5 \cdot 10^{-6}F)} \qquad = 65,89\ \text{Hz}$$

Potete quindi vedere come questo valore dipenda dalla tolleranza, cioè dai valori reali dei componenenti passivi che si impiegano.

Se riprendiamo la forma d'onda digitale che abbiamo appena discusso,

diventa più facile il funzionamento del circuito che mostriamo di seguito. Il monostabile 74121 fornisce l'impulso di gating mentre l'astabile 555 e l'Outboard di clock forniscono la sorgente di impulsi di clock.

Massa
GENERATORE DI IMPULSI DUALE
74121
C_1
30 kΩ
+ 5V
7400
Al pin 14 del contatore 7490 N° 1
1 kΩ
6.8 kΩ
555
C_2
CLOCK
nulla
Massa

Il timer 555 può essere usato con una alimentazione fino a +15 V. Per altri particolari e per altri usi, più insoliti, del chip 555, vi rimandiamo al libro sulle applicazioni del timer 555 di Howard Berlin pubblicato anch'esso dalla Jackson Italiana Editrice s.rl.

INTRODUZIONE AGLI ESPERIMENTI

I seguenti esperimenti servono a provare le caratteristiche funzionali dei monostabili 74121 e 555 e dell'astabile 555.

Esperimento N.	Commento
1	Misurare la durata dell'impulso di uscita del monostabile 74121, con l'ausilio della frequenza di 750 kHz del microcomputer MMD-1.
2	Misurare la frequenza dell'Outboard di clock con un impulso di durata pari a quella del precedente esperimento.
3	Misurare la frequenza del multivibratore astabile 555 con un impulso della durata pari a quella dell'Esperimento N. 1.
4	Mostrare come si usa il chip 555 come multivibratore monostabile. Il monostabile è sottoposto a trigger dal monostabile 74123.

ESPERIMENTO N. 1

Scopo

Lo scopo di questo esperimento è di misurare la durata dell'impulso prodotto da un chip monostabile 74121.

Configurazione dei pin del circuito integrato

Schema del circuito

Al pin clear del contatore

+5V

Massa

R_{ext}

C_{ext}

3 14 5 7

11

10

4

6

1

Q

$\bar{Q}$

74121

ϕ_2

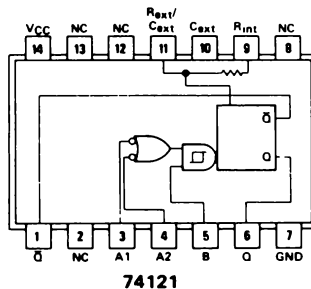
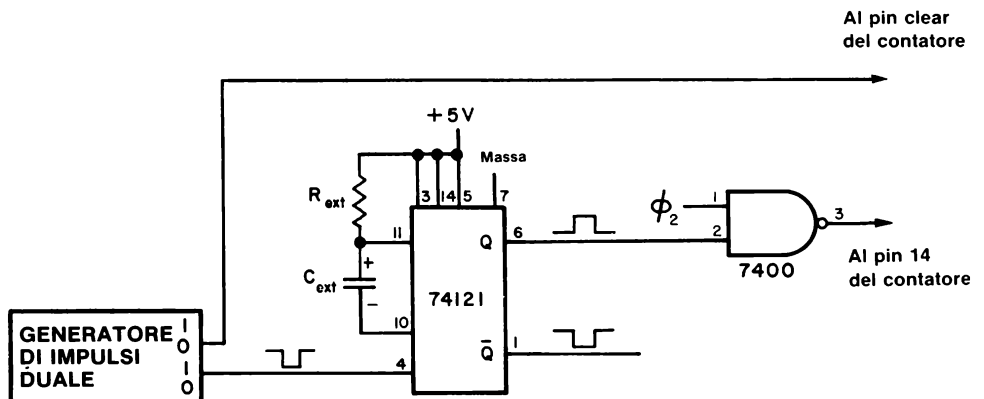
1

2

3

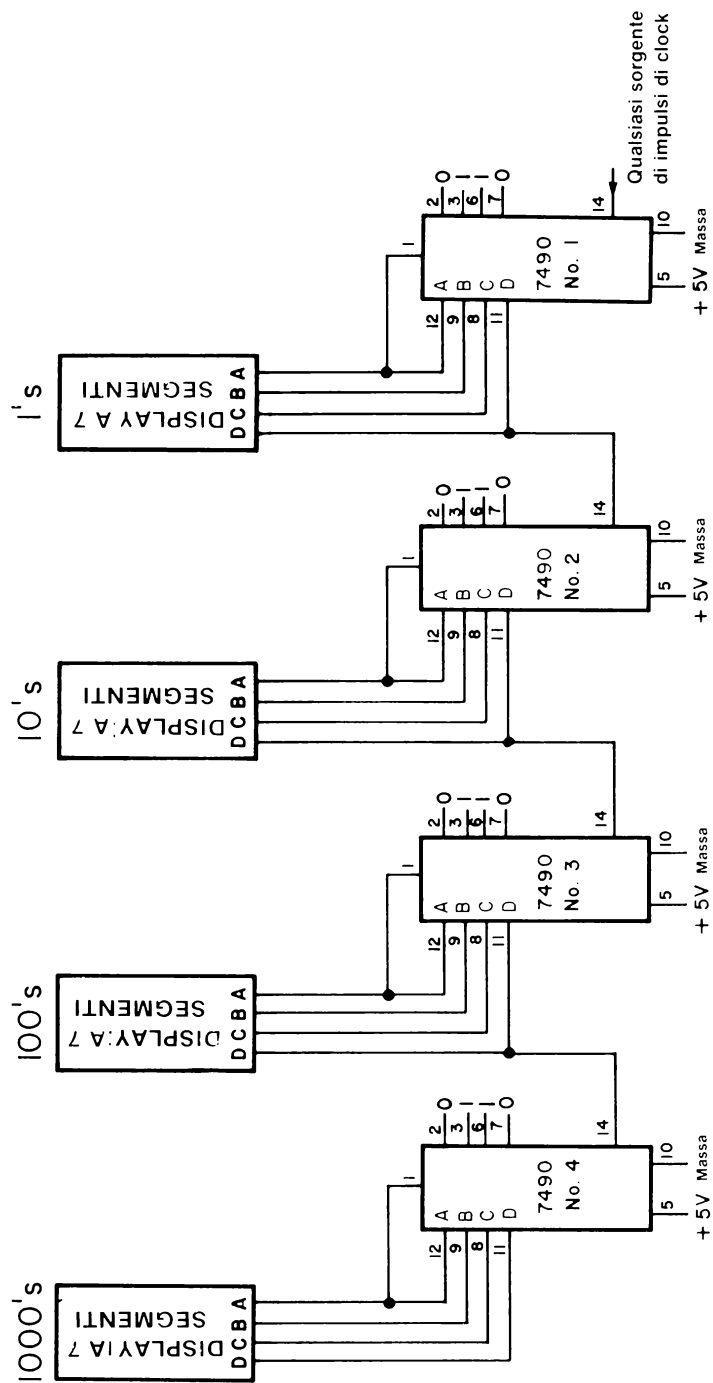
7400

Al pin 14 del contatore

GENERATORE DI IMPULSI DUALE

I O I O

Passo 1

Per questo esperimento avete bisogno di un contatore a più decadi, ma probabilmente avete già collegato il circuito del contatore a quattro decadi che mostriamo di seguito. Collegate gli ingressi clear del pin 2 tra loro e poi collegate il filo di connessione allo 0 logico o al generatore di impulsi. Siccome dovete fare alcune misurazioni, vi conviene avere un meccanismo che possa azzerare il contatore.

15-16

1000's
100's
10's
1's
DISPLAY A 7
SEGMENTI
7490
No. 4
7490
No. 3
7490
No. 2
7490
No. 1
+ 5V
Massa
Qualsiasi sorgente
di impulsi di clock

Passo 2

Collegate il circuito monostabile mostrato nello schema. La frequenza di clock Ø2 viene prelevata al pin di breadboard BP25 della piastra dell'MMD-1 alla sinistra del circuito integrato 8224. La frequenza a questo pin è di 750 kHz, e la stabilità è probabilmente migliore di 10 ppm e voi la userete per calibrare il vostro circuito monostabile.

Ricordate che probabilmente state usando dei resistori o dei condensatori che hanno una tolleranza del 10%, quindi, ogni durata di impulso che è all'interno di questa tolleranza, è accettabile. In realtà però, l'errore nella durata dell'impulso sarà più grande del 10% perchè sia i due resistori che il condensatore hanno singolarmente questa tolleranza. Che poi usate dei condensatori elettrolitici, l'errore sarà probabilmente più grande.

Passo 3

Prendete $R_{ext} = 10\ k\Omega$ e $C_{ext} = 0,1\mu F$. La durata dell'impulso dovrà essere calcolata usando la seguente equazione che abbiamo già dato,

$$t_{\omega} = R_{ext}\, C_{ext} \ln 2$$

dove t_{ω} = larghezza dell'impulso, secondi

R_{ext} = resistenza di timing esterno, ohm

C_{ext} = capacità di timing esterna, Farad

$\ln 2$ = logaritmo naturale di 2 = 0,693

Probabilmente la ripetitività del chip 74121 non è migliore di ±0,5%. Con i valori teorici di R_{ext} e C_{ext} che abbiamo detto, la durata dell'impulso dovrebbe essere,

$$t_{\omega} = 693\ \mu s$$

Con un ingresso di clock di 750 kHz alla porta NAND a 2-ingressi, il numero di conteggi che appaiono sul contatore sono,

$$\text{Conteggi} = 750\,000 \cdot 0,000693 = 519 \pm 1$$

Passo 4

Misurate il numero di conteggi con i valori della resistenza e della capacità di timing che abbiamo dato nel passo precedente. Quanti conteggi vengono effettuati? Ripetete la misurazione almeno dieci volte nell'arco di cinque minuti.

Noi abbiamo contato 481 conteggi che però variano da 475 a 483 nell'arco di tredici minuti, quindi l'errore delle nostre misurazioni, riferito al calcolo teorico di 519 conteggi, è del 7,3%, cioè all'interno della tolleranza dei componenti passivi.

Passo 5

Misurare il numero dei conteggi usando diversi condensatori, e mettere i risultati nella seguente tabella. Tenete fissa la R_{ext} di 10 kΩ.

Capacità μF	Conteggi teorici	Conteggi misurati	Errore in percentuale
0,02	104		
0,05	260		
0,10	519		
0,33	1716		
1	5199		
2,2	11,437		
3,3	17,155		

Noi abbiamo ottenuto i seguenti risultati. Abbiamo eseguito molte misurazioni per evidenziare gli scostamenti ottenuti.

Capacità μF	Conteggi misurati
0,02	92, 93, 92, 93, 92,92, 92
0,05	185, 185, 186, 187, 187, 187
0,10	481, 481, 482, 483, ... 477, 477, 475
0,33	2545, 2539, 2538, 2527, 2516, 2517, 2523, 2518, 2525
1	12.418; 16.909; 17.408; 16.979; 18.618; 19.120;
3,3	22.317; 21.994; 21.909; 21.887; 21.893; 21.790; 21.760; 21.788; 21.940

Il condensatore da 1 μF è considerevolmente in errore ed è lo stesso che abbiamo usato nell'Esperimento N. 7 del capitolo 13. Queste misurazioni sono sufficienti per darci una ragione per scartarlo, e lo facciamo.

Passo 6

Qual'è la durata dell'impulso che avete misurato con la capacità di 0,1 e 0,33 μF?

Noi abbiamo calcolato una durata d'impulso pari a 0,641 ms e 3,36 ms.

Passo 7

Quale sarebbe la durata dell'impulso teorica se C_{ext} = 50 pF e R_{ext} = 10 kΩ?

Circa 347 ns.

Se usate un impulso di gating di questo tipo per attuare il gate di una sorgente di impulsi di clock di 750 kHz per il vostro contatore, quanti conteggi osservereste?

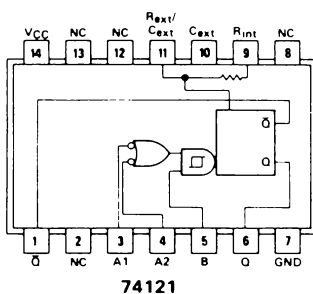
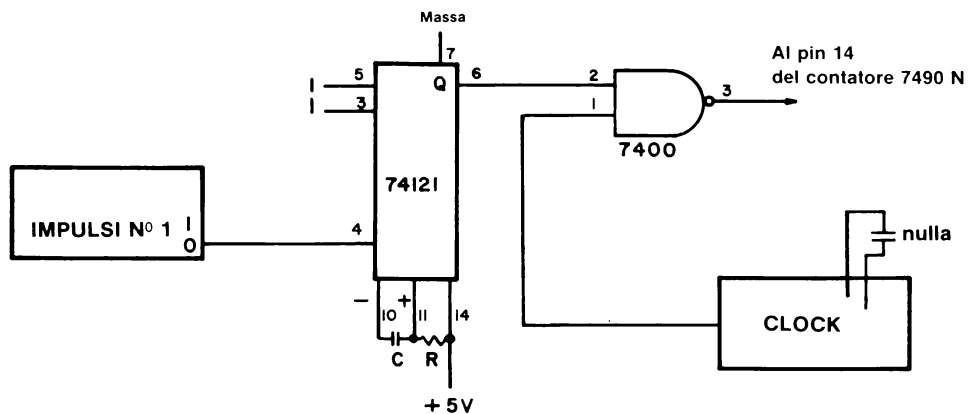
O nessun conteggio o 1 conteggio; dipende se arriva un fronte negativo all'uscita della porta NAND a 2-ingressi o no. Facendo una prova su cinquanta impulsi, noi abbiamo ottenuto undici volte nessun conteggio e trentanove volte un conteggio.

Conservate questo circuito e continuate col prossimo esperimento.

ESPERIMENTO N. 2

Scopo

Lo scopo di questo esperimento è di mostrare la frequenza dell'Outboard di clock senza alcuna capacità di timing.

Configurazione dei pin del circuito

Schema del circuito integrato

Passo 1

Fate le modifiche del circuito mostrato nello schema. La sola differenza è che bisogna sostituire l'Outboard di clock col clock da 750 kHz che si trova sulla piastra del l'MMD-1. I collegamenti col chip 74121 non cambiano.

Passo 2

Mettete il condensatore da 0,33 μF, lasciando la $R_{ext} = 10000\ \Omega$, e controllate la larghezza dell'impulso già misurata nell'Esperimento N. 1 e scrivetela qui sotto.

Noi abbiamo misurato una durata di 3,36 ms.

Passo 3

Determinate il numero di conteggi prodotto dall'Outboard di clock con l'impulso di questa durata e scrivete qui sotto il risultato. Ripetete la misurazione più volte.

Noi abbiamo ottenuto i seguenti risultati: 254 conteggi, 250, 249, 250, 250, 250, 250, e 247. Il valore medio è quindi 250.

Passo 4

Basandovi sulle osservazioni che avete fatto e conoscendo la durata dell'impulso del monostabile, calcolate la frequenza del vostro Outboard di clock.

Noi abbiamo calcolato la seguente frequenza:

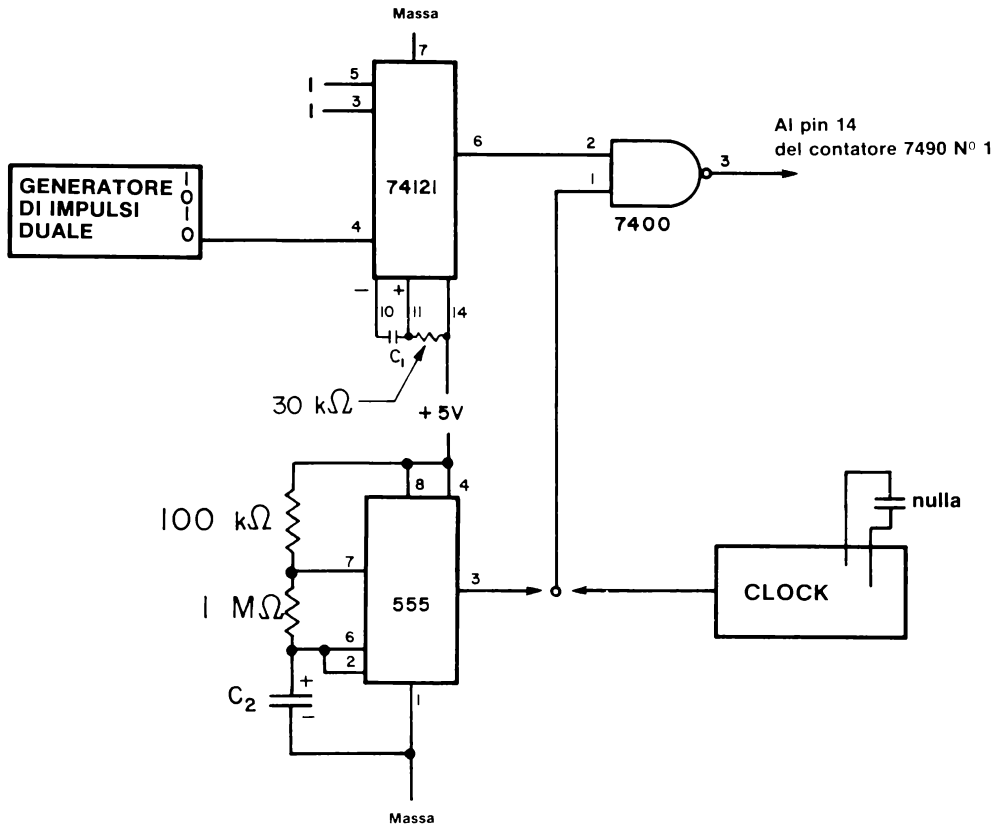
$$\text{Frequenza} = \frac{\text{Numero di conteggi}}{\text{Durata dell'impulso}} = 250 \text{ conteggi}/0{,}00336 \text{ s} = 74{,}4 \text{ kHz}$$

Conservate questo circuito e continuate col prossimo esperimento.

ESPERIMENTO N. 3

Scopo

Lo scopo di questo esperimento è di collegare il timer 555 come multivibratore astabile e di misurare la sua frequenza.

Schema del circuito

Configurazione dei pin del circuito integrato

Passo 1

In questo esperimento collegherete un clock direttamente dal timer 555. Le resistenze di timing sono identiche a quelle dell'Outboard di clock.

Collegate il circuito e alimentate il breadboard mettendo C_2=0 (capacità 0).

Passo 2

Usate la stessa durata d'impulso che avete usato per il chip 74121 nell'Esperimento N. 2. Con questa durata d'impulso determinate il numero di conteggi che si hanno col multivibratore astabile 555 come sorgente di impulsi di clock.

Noi abbiamo ottenuto i seguenti risultati: 291, 289, 289, 288, 289, 288, 288, 289, 287, 288, 288, 288, 287, 288, 288, 287, 288, 288, 287, 287 e 288 conteggi.

Passo 3

Calcolate la frequenza del multivibratore astabile 555.

Noi abbiamo ottenuto una frequenza pari a 85,7 kHz. *Conservate questo circuito.*

ESPERIMENTO N. 4

Scopo

Lo scopo di questo esperimento è di mostrare l'uso del timer 555 come multivibratore monostabile.

Schema del circuito

Configurazione dei pin del circuito integrato

Passo 1

Il multivibratore monostabile richiede in ingresso un impulso di trigger più corto dell'impulso in uscita, quindi gli impulsi validi sono compresi tra circa 2 μs e alcuni secondi. In questo esperimento userete un monostabile 74123 per fornire l'impulso di trigger al monostabile 555.

Passo 2

Collegate il circuito usando i seguenti componenti passivi:

monostabile 74123: C = nessuna

R = 1000 Ω

monostabile 555: C = 0,33 μF

R_A = 100 kΩ

L'outboard di clock dovrebbe essere usato come sorgente di impulsi di clock. Collegate il pin 4 del chip 74123 al pin 2 del chip 555; non usate il generatore di impulsi indicato nello schema del circuito monostabile 555; lo userete più tardi.

Passo 3

L'aspetto più interessante di questo esperimento è l'uso del monostabile per attuare il gate degli impulsi di clock che partono dal 555 instabile (l'Outboard di clock) e che vanno al contatore a più decadi. Sia il 555 monostabile che astabile hanno un alto grado di stabilità, così potrete osservare dei risultati migliori rispetto agli esperimenti precedenti.

Controllate il numero di conteggi quando date un trigger al monostabile 74123 e scrivete qui sotto i risultati che ottenete ripetendo più volte l'esperimento.

Quando abbiamo fatto l'esperimento abbiamo ottenuto i seguenti risultati: 7089 conteggi, 6808, 6782, 6682, 6649, 6606, 6584. 6586, 6568, 6550, 6526 e 6639. Dopo alcuni minuti abbiamo ottenuto : 6786, 6680, 6804, 6708, 6639, 6584 e 6600. È sorprendente come non si riesca a ottenere due valori uguali; ci saremmo aspettati di meglio dal monostabile 555.

Passo 4

Usate la frequenza dell'Outboard di clock, che avete determinato nell'Esperimento N. 2, per calcolare la durata dell'impulso del vostro monostabile 555. Scrivete qui sotto i calcoli che fate.

la frequenza dell'Outboard di clock era di 74,4 kHz, quindi la durata dell'impulso è:

$$t_\omega = \frac{\text{Numero di conteggi}}{\text{Frequenza}} = 6600 \text{ conteggi}/74400 \text{ Hz} = 88{,}7 \text{ ms.}$$

La durata teorica dell'impulso è:

$$t_\omega = 1{,}1\ R_A\ C = (1{,}1)\ (100000\ \Omega)\ (0{,}33\ \mu F) = 36{,}3 \text{ ms}$$

Non concordano molto. La formula teorica è stata presa dal Digital/linear/MOS Application Manual della Signetics Corporation.

Passo 5

Adesso togliete il circuito 74123 e usate il generatore di impulsi per fornire il trigger al monostabile 555. Azzerate il contatore, premete il generatore di impulsi e tenetelo premuto; dovreste vedere che il contatore continua a contare. Perchè? Perchè la durata dell'impulso di trigger è maggiore dell'impulso che si vorrebbe ottenere in uscita. Tutte le volte che si è in un caso di questo tipo, succede così.

DOMANDE RIEPILOGATIVE

Le seguenti domande servono ad aiutarvi a ripassare l'uso dei monostabili per attuare il gate di segnali di clock per un contatore.

1. Calcolate la durata dell'impulso di uscita di un monostabile 74121 con i seguenti valori di resistore e condensatore segnatempo. Usate la formula $t_\omega = R_{ext} C_{ext} \ln 2$

	R_{ext}, Ω	C_{ext}, μF
a.	2000	0,0005
b.	5000	0,001
c.	30000	0,01
d.	30000	0,1
e.	30000	1,0

2. Calcolate la frequenza dell'impulso del monostabile dati i seguenti valori di frequenza di clock e di conteggi effettuati.

	Frequenza di clock kHz	Conteggi
a.	750	15
b.	750	67238
c.	750	789002
d.	74,4	62389
e.	74,4	4333
f.	74,4	167

3. Se la frequenza di clock fosse 750 kHz, quanti conteggi misurereste su un contatore per ognuno dei seguenti valori di durata di impulso?

   a. 50 ns

   b. 500 ns

   c. 1,333 μs

   d. 50 μs

   e. 500 μs

   f. 50 ms

   g. 1 ms

RISPOSTE

1. a. 693 ns
   b. 3,47 μs
   c. 208 μs
   d. 2,08 ms
   e. 20,8 ms

2. a. 20,0 ms
   b. 89,7 ms
   c. 1,05 s
   d. 839 ms
   e. 58,2 ms
   f. 2,23 ms

3. a. 0 o 1 conteggio
   b. 0 o 1 conteggio
   c. 1 conteggio
   d. 37 o 38 conteggi
   e. 375 conteggi
   f. 37500 conteggi
   g. 750 conteggi

APPENDICE 1

RIFERIMENTI

1. *The Compact Edition of the Oxford English Dictionary,* Oxford Univ. Press, 1971.

2. Rudolf F. Graf, *Modern Dictionary of Electronics,* Howard W. Sams & Company, Inc., Indianapolis, 1972.

3. James Martin, *Telecommunications and the Computer,* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1969.

4. Abraham Marcus and John D. Lenk, *Computers for Technicians,* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1973.

5. Microdata Corporation, *Microprogramming Handbook,* Santa Ana, California, 1971.

6. J. Blukis and M. Baker, *Practical Digital Electronics,* Hewlett-Packard Company, Santa Clara, California, 1974.

7. Donald E. Lancaster, *TTL Cookbook,* Howard W. Sams & Co., Inc., Indianapolis, 1974

8. H. V. Malmstadt, C. G. Enke, and S. R. Crouch, *Instrumentation for Scientists Series, Module 3. Digital and Analog Data Conversion,* W. A. Benjamin, Inc., Menlo Park, California, 1973-4.

9. H. V. Malmstadt and C. G. Enke, *Digital Electronics for Scientists,* W. A. Benjamin, Inc., New York, 1969.

10. J.D. Lenk, *Handbook of Logic Circuits,* Reston Publishing Company, Inc., Reston, Virginia, 1972.

11. A. James Diefenderfer, *Principles of Electronic Instrumentation,* W. B. Saunders Company, Philadelphia, 1972.

12. P. R. Rony and D. G. Larsen, *Bugbook II. Logic & Memory Experiments Using TTL Integrated Circuits,* E & L Instruments, Inc., Derby, Connecticut, 1974.

13. Robert L. Morris and John R. Miller, Editors, *Designing with TTL Integrated Circuits,* Mc Graw-Hill Book Company, New York, 1971.

14. Charles J. Sippl, *Microcomputer Dictionary and Guide,* Matrix Publishers, Inc., Champagne, Illinois 61820, 1976.

15. Donald Eadie, *Introduction to the Basic Computer,* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1973.

16. Texas Instruments Incorporated, *Microprocessor Handbook,* Dallas, Texas, 1975.

17. Charles L. Garfinkel of Keithley Instruments, Inc., is the originator of this definition.

APPENDICE 2

DIZIONARIO DEI TERMINI TECNICI DEL BUGBOOK V

Questa appendice costituisce un piccolo dizionario dei termini tecnici più importanti relativi all'elettronica digitale e ai microcomputer riportati nel Bugbook V. Per una migliore consultazione i termini sono indicati sia in italiano che in inglese (fra parentesi).
Le definizioni sono state tratte dai seguenti testi:

- Rudolf F. Graf, *Modern Dictionary of Electronics,* Howard W. Sams & Co., Inc., Indianapolis, 1972.
- Microdata Corporation, *Microprogramming Handbook,* Santa Ana, California, 1972.
- Donald Eadie, *Introduction to the Basic Computer,* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1973.
- Abraham Marcus and John D. Lenk, *Computers for Technicians,* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1973.
- Peter R. Rony, David G. Larsen, and Jonathan A. Titus, *Bugbook III. Microcomputer Interfacing Experiments Using the Mark 80 Microcomputer, an 8080 System,* E & L Instruments, Inc., Derby, Connecticut, 1975.

***

Abilitare
(To enable)
Permettere il passaggio di un segnale digitale in o attraverso un dispositivo digitale o un circuito. Pag. 11-12.

Accumulatore
(Accumulator)
Il registro e la associata circuiteria elettronica digitale nell'unità aritmetico/logica (ALU) di un computer, in cui vengono realizzate le operazioni logiche ed aritmetiche. Pag. 3-7.

Algebra Booleana
(Boolean algebra)
Un sistema di logica matematica relativo alle classi, alle proporzioni, elementi circuitali on-off, etc., associati da operatori quali l'AND, OR, NOT, XOR, etc. Il nome deriva dal matematico inglese George Boole che introdusse questa logica nel 1847. Pag. 8-2.

Base
(Base)
Indicata anche con il nome di radice. Il numero totale dei diversi simboli utilizzati in un dato sistema di numerazione. Ad esempio, dato che il sistema di numerazione decimale utilizza dieci simboli, la radice è 10. Nel sistema di numerazione ottale, la radice è 8. Nel sistema di numerazione binario, la radice è 2 perchè si hanno solo due simboli (0 e 1). Pag. 1-4.

Binario
(Binary)

Sistema di numerazione utilizzante una base, o radice, due. Nel sistema binario vi sono due digit (0 e 1). Pag. 7-2.

Binario decimale codificato
(Binary coded decimal)

L'abbreviazione è BCD. Si tratta di un sistema di rappresentazione dei numeri in cui ogni digit decimale di un dato numero è espresso da numeri binari.
Conosciuto anche come codice 8421. Pag. 12-4.

Bit
(Bit)

Abbreviazione per *BI*nary digi*T*. Unità di informazione eguale all'uno binario, oppure indicazione di uno dei due stati (0 e 1) utilizzati per memorizzare o trasferire una informazione. Pag. 1-3.

Breadboard
(Breadboard)

Qualsiasi ausilio utilizzato per collegare in modo temporaneo tra loro elementi circuitali, al fine di verificare la funzionalità di un dato circuito. Di solito si tratta di piastre o basette recanti componenti o circuiti. Pag. 9-2

Breadboarding
(Breadboarding)

Azione relativa all'utilizzo di un breadboard per attuare dei collegamenti temporanei in circuiti elettrici. Pag. 9-2

Buffer
(Buffer)

Elemento di un circuito digitale che può essere utilizzato per gestire un alto fan-out oppure per invertire livelli di ingresso/uscita. Pag. 14-20

Buffer gate
(Buffer gate)

Circuito digitale che incrementa la capacità di fornire tensione o corrente da parte di un circuito binario. Conosciuto anche come driver (pilota). Pag. 7-13

Byte
(Byte)

Gruppo di 8 bit contigui considerati come unità ed occupanti una singola locazione di memoria. Pag. 2-6

Byte di dati
(Data byte)

Per un microcomputer basato sull'8080, un byte di dati è un numero binario ad 8 bit trasferito sul bus dati bidirezionale. Pagg. 3-3 e 3-5

Byte di indirizzo LO
(LO address byte)

Gli 8 bit meno significativi di un indirizzo di memoria a 16 bit, per il microprocessor 8080. Pagg. 2-8 e 3-5

Capacità
(Capacitance)

È la capacità di un condensatore o di un sistema conduttore/dielettrico di immagazzinare cariche elettricamente separate quando esiste una differenza di potenziale tra i conduttori. La capacità di un condensatore è definita come il rapporto tra la carica elettrica che è stata trasferita da un elettrodo ad un altro e la differenza di potenziale risultante tra i due elettrodi. L'unità di misura della capacità è il Farad. Pag. 9-11. C (Farads) = Q (Coulombs) / V (Volt).

Circuito di gate
(Gate circuit)

Circuito che permette il passaggio di un segnale solo quando è presente un impulso di gating. Circuito elettronico con uno o più ingressi e un'uscita in cui un impulso esce sulla linea di uscita se, e solo se, si verificano delle specifiche combinazioni di impulsi alle linee di ingresso. Pag. 14-20

Circuito di gating (Gating circuit)	Circuito che opera come switch selettivo e permette la conduzione solo durante selezionati intervalli di tempo oppure quando la grandezza del segnale è entro specifici limiti. Pag. 14-21
Clear (Clear)	Vedere reset. Pag. 11-7
Clock (Clock)	Detto anche orologio. Qualunque dispositivo che genera uno o più impulsi di clock. Pag. 9-12
Clock (Clock)	Un generatore di impulsi che controlla la temporizzazione di dispositivi "clocked logic" e che regola la velocità cui tali dispositivi operano. Svolge anche un'azione di sincronizzazione tra tutte le operazioni in un sistema digitale. Pag. 11-5
Codice ASCII (ASCII code)	American Standard Code for Information Interchange. Codice a sette bit, senza bit di parità, oppure ad otto se con parità. Pag. 12-6
Codice binario (Binary code)	Codice in cui ogni elemento distingue uno dei due stati. Questi stati sono indicati con i simboli 0 ed 1. Pag. 1-4
Codice di dispositivo (Device code)	In un microcomputer basato sull'8080, rappresenta un codice ad 8 bit per uno specifico dispositivo di ingresso/uscita. Pagg. 3-3 e 3-5
Codice macchina (Machine code)	Rappresentazione binaria di un'istruzione per computer. Pag. 2-4
Codice mnemonico (Mnemonic code)	Istruzione scritta in una forma tale da facilitare la comprensione (simbolicamente), ma che può essere convertita in codice macchina. Pag. 2-4
Codice operazione (Operation code)	Per un microprocessor 8080, il codice ad 8 bit relativo ad una specifica azione. Pag. 3-5
Codice ottale (Octal code)	Relativo al sistema di numerazione binario a base 8, in cui i numeri da 0 a 7 sono usati per rappresentare i digit ottali da 0 a 7. Pag. 1-5
Codificare (To code)	Utilizzare un codice ,spesso costituito da numeri binari, per rappresentare dei singoli caratteri o gruppi di caratteri nell'ambito di un messaggio. Effettuare un cambiamento da un codice ad un altro. Se i codici sono molto differenti, il processo è detto conversione di codice. Pagg. 1-5 e 12-6
Complemento (Complement)	Complemento di un numero binario. Il complemento di 1 è 0 e il complemento di 0 è 1. Il complemento di 011010 è 100101. Pag. 8-7
Computer (Computer)	Vedere computer digitale. Pag 2-2

Computer digitale (Digital computer)	Un dispositivo elettronico in grado di accettare, memorizzare, manipolare da un punto di vista aritmetico delle informazioni, intese sia come dati che come istruzioni. Le informazioni sono trattate in termini di digit codificati secondo il sistema di numerazione binario (0 e 1), e sono rappresentati da due livelli di tensione. Pag. 2-2
Comunicazione (Communication)	Fornire, inviare, scambiare idee, conoscenze ed informazioni, tramite parole, scritti, segni o segnali. Pag. 1-2
Condensatore (Capacitor)	Componente elettronico consistente in due superfici conduttrici separate da materiale isolante e dielettrico (carta, mica, film plastico, dielettrico inorganico). Un condensatore immagazzina energia elettrica, blocca il flusso di corrente continua e permette il flusso di corrente alternata. Tutto ciò essenzialmente in funzione del valore della capacità e della frequenza. Pag. 9-11
Contatore (Counter)	Dispositivo in grado di cambiare stato in una specifica sequenza in base al ricevimento di appropriati segnali di ingresso. L'uscita del contatore indica il numero degli impulsi che sono stati applicati (vedere anche divisore). Un contatore deriva dall'insieme di flip-flop con alcune porte (gate). L'uscita di tutti i flip-flop è accessibile per indicare l'esatto conteggio in ogni istante. Pag. 13-2
Contatore a decade (Decade counter)	Dispositivo logico con 10 stati stabili, in grado di ciclare attraverso questi stati tramite l'applicazione di dieci impulsi di clock. Un contatore a decade usualmente conta in una sequenza binaria dallo stato 0 allo stato 9 e successivamente ritorna allo stato 0. È anche indicato come contatore-divisore per 10. Pag. A-31
Contatore binario (Binary counter)	Interconnessione di flip-flop avente un singolo ingresso, realizzata in modo da permettere un conteggio binario. Ogni volta che un impulso appare all'ingresso, il contatore cambia stato: il numero degli impulsi in ingresso è tabulato in modo da fornire una informazione in uscita in forma binaria. I conteggi possibili sono 2n, dove n è il numero dei flip-flop o degli stadi. Pag. 13-2
Conversione di codice (Code conversion)	Cambiamento della configurazione di bit per un carattere in un codice, nella corrispondente configurazione di bit in un altro codice. Pag. 12-8
Corsa (Race)	Condizione che si verifica quando il cambiare lo stato di un sistema richiede un cambiamento in due o più stati variabili. Se lo stato finale subisce un'influenza dallo stato variabile che per primo cambia, la condizione è detta corsa critica. Anche la condizione che esiste quando un segnale si propaga attraverso due o più elementi di memoria durante lo stesso periodo di clock. Pag. 11-11

Decodificare (To decode)	Utilizzo di un codice per modificare una precedente codifica. Determinare il significato di un set di impulsi o di segnali logici utilizzati per descrivere una istruzione, un comando od una operazione. Pag. 12-6
Diodo (Diode)	Semiconduttore a due elettrodi utilizzante le capacità raddrizzatrici di una giunzione pn (diodo a giunzione) oppure di un punto metallico in contatto con un diodo a semiconduttore (diodo a punto di contatto - point contatct diode). Pag. 9-12
Disabilitare (To disable)	Impedire il passaggio di segnali digitali tramite l'applicazione di un opportuno segnale al terminale di disabilitazione di un dispositivo digitale. Pagg. 11-2 e 14-20
Display (Display)	Detto anche indicatore. È un dispositivo che permette di visualizzare un segnale elettronico. Pag. A-24
Dispositivo digitale (Digital device)	Qualsiasi dispositivo che opera o manipola informazioni binarie. Pag. 7-2
Driver (Driver)	Detto anche pilota. Elemento di un circuito digitale che viene accoppiato allo stadio di uscita di un circuito per aumentare (pilotare) la capacità di controllo in tensione, in corrente o di fan-out. Per esempio un clock driver viene utilizzato per fornire la necessaria corrente alla linea di clock. Vedere Buffer Gate. Pag. 14-21
Edge triggered flip-flop (Edge-triggered flip-flop)	Tipo di flip-flop in cui una minima percentuale di variazione del segnale di clock, in volt/secondi, è una condizione necessaria per determinare il cambiamento dell'uscita. Pag. 13-6
Elemento astabile (Astable element)	Elemento a due stati che non ha uno stato stabile. Pag. 15-2
Elemento bistabile (Bistable element)	Differente terminologia per indicare il flip-flop. Circuito in cui l'uscita ha due stati stabili raggiungibili in base agli ingressi. Lo stato in uscita rimane anche all'esaurirsi del segnale in ingresso. Pagg. 11-2 e 15-2
Fan-in (Fan-in)	Il carico richiesto da un ingresso digitale ad un circuito integrato. Per la famiglia TTL, il carico di ingresso richiesto è normalizzato al valore 1. Un fan-in di 1 corrisponde a 1,6 mA. Pag. 10-16
Fan-out (Fan-out)	Il numero di carichi paralleli nell'ambito di una certa famiglia logica, come la TTL, che possono essere pilotati dall'uscita di un circuito logico. Un chip TTL standard ha un fan-out di 10 il che significa che può pilotare 10 carichi TTL standard ciascuno con un fan-in di 1. Un fan-out di 10 nel caso TTL corrisponde a 16 mA. Pagg. 10-16 e 14-21

Flip-Flop (Flip-flop)	Un circuito avente due stati stabili e le capacità di spostarsi da uno stato ad un altro tramite l'applicazione di un segnale di controllo; il flip-flop resta in questo stato anche dopo che il segnale è stato rimosso. Pag. 11-12
Flip-flop di tipo D (D-type flip-flop)	Il simbolo D indica Delay (ritardo). È un flip-flop la cui uscita è in ritardo rispetto all'ingresso di un impulso di clock; ad esempio, se un 1 logico appare all'ingresso, l'uscita diventerà 1 dopo il successivo impulso di clock. Pag. 11-5
Forma d'onda digitale (Digital waveform)	Rappresentazione grafica di un segnale digitale, indicante le variazioni dello stato logico in funzione del tempo. Questo tipo di rappresentazione e anche conosciuto con il nome di diagramma di tempo (timing diagram). Pag. 11-9
Fronte negativo (Negative edge)	La transizione dall'1 logico allo 0 logico in un impulso di clock. Pag. 13-6
Fronte positivo (Positive edge)	La transizione dallo 0 all'1 logico in un impulso di clock. Pag. 13-6
Funzione ausiliaria (Auxiliary function)	Tutti i diversi dispositivi elettronici necessari per rendere operativo un circuito digitale: circuiti integrati, resistori, condensatori, etc. Pag. 9-4
Gated buffer (Gated buffer)	Un circuito driver a bassa impedenza che può essere usato per pilotare una linea per differenziazione di impulsi oppure in multivibratori. In generale si dice di un buffer che è "gated". Pag. 14-20
Gate (dispositivo di gating) [*Gate (gating device)*]	Circuito avente due o più ingressi e un'uscita. Uno degli ingressi può essere identificato come ingresso dati ed i rimanenti come ingresso di gating. Lo stato logico degli ingressi di gating determina il passaggio o meno dell'ingresso dati che appare all'uscita. Pag. 14-6
Gate (dispositivo logico) [*Gate (logic device)*]	Circuito con due o più ingressi e un'uscita. L'uscita dipende dalla combinazione dei segnali logici in ingresso. Vi sono 4 porte (gate) base AND, OR, NAND, NOR. Pagg. 7-2, 14-6 e 14-15
Gated driver (Gated driver)	In generale, un driver che è un "gated". Pag. 14-21
Generatore di impulsi (Pulser)	Switch logico che genera un singolo impulso di clock. Pag. 9-12
Impulso di clock (Clock pulse)	Un ciclo logico completo dallo 0 logico all'1 logico ed ancora allo 0 logico (impulso di clock positivi), oppure dall'1 logico allo 0 logico ed ancora all'1 logico (impulso di clock negativo). Pag. A-27

Impulso di gate
(Gate pulse)

Impulso che abilita (enable) un circuito di gate (gate circuit) a far passare un segnale. L'impulso di porta generalmente ha una durata più lunga del segnale per assicurare la coincidenza. Pag. 14-20

Impulso di gating
(Gating pulse)

Impulso che modifica o controlla l'operazione di un circuito di gate (gate circuit). Pag. 14-21

Indicatore a LED
(LED lamp monitor)

Diodi ad emissione di luce (LED), accesi nello stato logico 1, e spenti nello stato logico 0. Pag. A-23

Indirizzo di memoria
(Memory address)

La locazione di memorizzazione di una parola di memoria. Pag. 2-7

Ingressi asincroni
(Asynchronous inputs)

Quei pin di ingresso in un flip-flop che possono influire sullo stato di uscita indipendentemente dal clock; sono indicati con il nome di preset, reset e clear. Pag. 11-5

Ingressi sincroni
(Synchronous inputs)

Quei terminali di un flip-flop attraverso cui i dati possono entrare ma solo a seguito del comando di un clock. Questi ingressi che non hanno un controllo diretto dell'uscita come quelli di una porta sono detti ingressi JK e D. Pag. 11-5

Ingresso di clock
(Clock input)

Quel terminale di un flip-flop il cui stato, o cambiamento di stato, controlla l'ingresso di dati in un flip-flop attraverso gli ingressi sincroni, con conseguente controllo dello stato di uscita del flip-flop. Il segnale di clock attua due funzioni: (1) permette l'ingresso dati al flip-flop; (2) dopo l'ingresso, dirige concordemente il cambiamento di stato del flip-flop. Pag. 11-5

Invertitore
(Inverter)

Dispositivo digitale che complementa un segnale d'ingresso digitale. Pag. 7-9

Istruzione a due byte
(Two byte instruction)

Istruzione consistente in una informazione che occupa due successive locazioni di memoria. Pag. 3-2

Istruzione a tre byte
(Three byte instruction)

Istruzione consistente in una informazione che occupa tre successive locazioni di memoria. Pag. 3-2

Istruzione in singolo byte
(Single-byte instruction)

Istruzione consistente in 8 bit contigui, occupante una sola locazione di memoria. Pag. 3-2

Istruzione logica
(Logical instruction)

Operazione logica che è attuata da una coppia di parole dati multi-bit, in cui i bit corrispondenti di ogni parola partecipano ad un'operazione logica a due bit, come AND, OR e OR esclusivo. Pag. 8-2

Istruzione mnemonica
(Mnemonic instruction)

Istruzione di un calcolatore scritta in notazione simbolica come: ADD, SUB, MOV, DIV, MPY, STO, DIV. Pag. 2-4

Latch
(Latch)

Semplice elemento di memorizzazione. Un loop di controreazione utilizzato in un circuito digitale simmetrico, come un flip-flop, per mantenere uno stato logico. Pag. 11-5

Leggere
(To read)

Trasmettere dati da una specifica locazione di memoria verso qualche altro dispositivo digitale. Pag. 16-15

Linguaggio (Language)	L'insieme delle parole e dei metodi di combinazione delle parole usate da una nazione, da un popolo o da una razza. Pag. 1-2
Linguaggio mnemonico (Mnemonic language)	Linguaggio di programmazione che è basato su simboli facilmente ricordabili e che può essere assemblato nel codice macchina di un computer. Pag. 2-4
Mascheramento (Masking)	Tecnica logica in cui certi bit di una parola multi-bit sono eliminati od inibiti. Pag. 8-13
Memoria (Memory)	Qualunque dispositivo che può memorizzare bit 0 ed 1 in modo tale che un singolo bit oppure un gruppo di bit può essere ripreso ed utilizzato. Pag. 2-6
Memoria a lettura/scrittura (Read/write memory)	Memoria a semiconduttore in cui possono essere scritti e letti stati logici 0 ed 1. Pag. 2-6
Memoria a sola lettura (Read-only memory)	Memoria a semiconduttore da cui dati digitali possono essere letti ma in cui non è possibile inserire dati. Pag. 2-7
Microcomputer (Microcomputer)	Un completo computer digitale basato su un microprocessor o una famiglia di microprocessor. Pag. 2-2
Modulo (Modulo)	Il numero dei diversi stati per cui passa un contatore, prima di ripetere da capo la sequenza. Pag. 13-2
Multiplexer (Multiplexer)	Dispositivo digitale che può selezionare uno o un numero di ingressi e trasmettere lo stato logico di tale ingresso all'uscita. Pag. A-31
Multivibratore monostabile (Monostable multivibrator)	Circuito digitale che ha un solo stato stabile, da cui può essere spinto a cambiare stato, ma solo per un definito intervallo di tempo, dopo di che ritorna allo stato originale. Detto anche multivibratore One-shot, single shot multivibrator, start-stop multivibrator. Pag. 15-2
Operazione (Operation)	Specifica azione che un computer attua in base ad un'istruzione; esempio addizione, sottrazione, OR, AND etc. Pag. 3-2
Operazione mnemonica (Mnemonic operation)	Vedere istruzione mnemonica. Pag. 2-4
Parola (Word)	Numero di bit che un computer può manipolare simultaneamente. Pag. 2-6
Pilota (Driver)	Vedere driver. Pag. 14-21
Porta (Gate)	Vedere gate.
Porta AND (AND gate)	Circuito binario con due o più ingressi ed una sola uscita. L'uscita è allo stato logico 0 se uno qualunque degli ingressi è allo stato logico 0; l'uscita è allo stato logico 1 se tutti gli ingressi sono allo stato logico 1. Pag. 7-6

Porta NAND (NAND gate)	Combinazione di una funzione NOT e una funzione AND in un circuito binario che ha due o più ingressi e una sola uscita. L'uscita è allo 0 logico solo se tutti gli ingressi sono all'1 logico. L'uscita è all'1 logico se uno qualunque degli ingressi è allo 0 logico. Pag. 7-8
Porta NOR (NOR gate)	Una porta OR seguita da un invertitore; l'uscita di tale circuito binario è allo 0 logico se uno qualunque degli ingressi è all'1 logico; è all'1 logico se tutti gli ingressi sono allo 0 logico. Pag. 7-10
Porta NOT (NOT gate)	Circuito binario con una sola uscita sempre allo stato logico opposto a quello dell'ingresso. Detto anche circuito invertitore. Pag. 7-9
Porta OR-esclusivo (Exclusive-OR gate)	Circuito binario con due ingressi ed una sola uscita in cui l'uscita è allo stato logico 1 quando gli ingressi sono a stati logici differenti. L'uscita è a 0 se entrambi gli ingressi sono allo stesso stato logico. Pag. 7-11
Preset (Preset)	Ingresso asincrono utilizzato per controllare lo stato logico dell'uscita Q di un flip-flop. I segnali in ingresso attraverso il preset, determinano il passaggio dell'uscita Q verso lo stato logico 1. L'ingresso di preset non può far apparire 0 all'uscita. Pag. 11-17
Programma per computer (Computer program)	Una sequenza di istruzioni che, prese come gruppo, permettono al computer di effettuare un dato compito. Pagg. 2-2 e 5-2
Radice (Radix)	Vedere base. Pag. 1-4
Registro (Register)	Circuito digitale elettronico di memorizzazione avente come capacità una parola oppure un byte. Pag. 3-7
Registro non specializzato (General purpose register)	In un microcomputer basato sull'8080, vi sono 6 registri ad 8 bit, utilizzati per memorizzare informazioni in modo temporaneo. I registri sono: B, C, D, E, H, L. Pag. 6-2
Reset (Reset)	Ingresso asincrono utilizzato per controllare lo stato logico dell'uscita Q di un flip-flop. I segnali in ingresso attraversano il reset, determinano il passaggio dell'uscita Q verso lo stato logico 0. L'ingresso di reset non può far apparire 1 all'uscita Q. Pag. 11-7
Resistenza (Resistance)	Proprietà dei conduttori che in funzione della loro dimensione, materiale e temperatura, determina la corrente prodotta da una data differenza di potenziale. Quella proprietà di una sostanza che ostacola il passaggio della corrente con conseguente dissipazione di potenza sotto forma di calore. L'unità di misura della resistenza è l'ohm. Quest'ultimo è definito come la resistenza attraverso cui una differenza di potenziale di un volt produce una corrente di un ampére. Pag. 9-11

Resistore (Resistor)	Componente elettronico che connesso in un circuito elettrico introduce una specifica resistenza. Pag. 9-11
Schema elettrico (Schematic diagram)	Rappresentazione grafica di circuiti elettronici. Pag. 9-10
Scrivere (To write)	Trasmettere dati da qualche dispositivo digitale. Verso una specifica locazione di memoria. Pag. 16-15
Segnale binario (Binary signal)	Tipicamente una corrente od una tensione che fornisce una informazione in termini di cambiamento tra due differenti stati: stato logico 0 e stato logico 1. Pag. 14-2
Segnale di gate (Gate signal)	Vedere impulso di porta. Segnale che permette ad un circuito di porta (gate circuit) di far passare un segnale. Pag. 14-20
Segnali digitali (Digital signals)	Segnali discreti o discontinui i cui vari stati sono intervalli discreti. Pag. 14-2
Set (Set)	Vedere preset. Pag. 11-7
Simbolo (Symbol)	Carattere scritto utilizzato per rappresentare qualunque cosa: lettere, figure, segni convenzionali indicanti un qualche oggetto, processi, etc.. Pag. 9-10
Simbolo Booleano (Boolean symbol)	Simbolo utilizzato per rappresentare una specifica operazione booleana. Pag. 8-2
Simbolo mnemonico (Mnemonic symbol)	Un simbolo scelto per aiutare la memoria umana: es. MPY è la abbreviazione di "multiply". Pag. 2-4
Sincrono (Synchronous)	Operazione di un sistema a clocked logic con l'ausilio di un generatore di impulsi di clock. Tutte le azioni sono sincrone con il clock. Pag. 16-17
Switch (Switch)	Dispositivo elettrico o meccanico che attiva o interrompe una linea interessata al passaggio di corrente, o che indirizza la corrente a differenti possibili vie. Pag. 14-21
Switch logico (Logical switch)	Dispositivo meccanico che presenta sia uno stato logico 0 che uno stato logico 1 al terminale di uscita. Pag. 9-12
Tabella della verità (Truth table)	Tabulazione che mostra le relazioni tra tutti i livelli logici di uscita di un circuito digitale in funzione di tutte le possibili combinazioni dei livelli logici in ingresso; da cui la completa caratterizzazione funzionale di un dato circuito: Pag. 7-3
Tempo di discesa (Fall time)	Il tempo necessario al fronte di discesa di un impulso per passare dal 90% al 10% del suo valore iniziale. In elettronica digitale, è il tempo necessario ad una tensione di uscita di un circuito digitale per cambiare da alto livello (1 logico) a basso livello di tensione (0 logico). Pag. 11-10

Tempo di salita
(Rise-time)

Il tempo richiesto dal fronte di salita di un impulso positivo per passare dal 10% al 90% del valore finale. È proporzionale alla costante di tempo ed è la misura della pendenza del fronte d'onda. In elettronica digitale, il tempo richiesto perchè una tensione in uscita da un circuito digitale cambi da basso livello (0 logico) ad alto livello (1 logico). Pag. 11-10

Tempo totale di ritardo di propagazione
(Propagation delay)

Detto anche delay di propagazione.
È il tempo richiesto da un segnale logico per passare attraverso una serie di dispositivi logici. Vi sono 4 tipi di ritardi circuitali: memorizzazione (storage) salita (rise), discesa (fall) e ritardo di turn-on. Tempo di ritardo totale tra quando un segnale d'ingresso attraversa il punto di soglia di un valore di tensione e quando la corrispondente uscita attraversa lo stesso punto di tensione. Pag. 11-10

Teorema di DeMorgan
(DeMorgan's theorem)

Teorema secondo cui l'inversione di una serie di AND è uguale alla stessa serie di OR singolarmente invertiti; oppure l'inversione di una serie di OR è uguale alla stessa serie di AND singolarmente invertiti; simbolicamente è:

$A \cdot B \cdot C = A + B + C$

$A + B + C = A \cdot B \cdot C$

To gate
(To gate)

Controllare il passaggio di un segnale digitale attraverso un circuito digitale. Pag. 15-15

To strobe
(To strobe)

Attivare un circuito digitale.
Vedere abilitare (enable). Pagg. 11-12 e 14-16

To trigger
(To trigger)

Fornire un impulso che fa partire una certa azione.
Può anche essere il fronte di un impulso. Vedere abilitare (enable). Pagg. 11-12, 13-6 e 14-21

XOR
(XOR)

Vedere porta OR-esclusivo. Pag. 7-11

APPENDICE 3

OUTBOARDS®

Outboard® è un marchio registrato della E-L Instruments, Inc. e indica un elemento, che svolge una funzione ausiliaria, montato su un piccolo circuito stampato adatto al collegamento diretto sulla piastra SK-10. I collegamenti +5 V e massa per gli Outboards sono realizzati attraverso le file di terminali senza saldature presenti sull'SK-10. Sono ottenuti dalle due file dei terminali. I pin di ingresso-uscita vengono elettricamente collegati ai sets di 5 terminali senza saldatura presenti nella parte interna della piastra SK-10. Le caratteristiche fisiche di un Outboard possono essere facilmente dedotte dalla figura A1, che rappresenta l'Outboard LR-7, dual pulser visto da sotto. I collegamenti con l'SK-10 avvengono in 6 punti: +5V (fila esterna del bus di terminali di alimentatione), 0V (parte interna del bus di terminali di alimentazione) e 4 terminali di ingresso. Per realizzare queste connessioni, dovete premere leggermente l'Outboard sulla piastra SK-10 nell'opportuna posizione.
I vantaggi del concetto di Outboard possono essere così riassunti: gli Outboard sono compatti, facilmente reperibili, economici, portatili e possono essere collocati ovunque sulla piastra SK-10. I collegamenti +5V e Massa per gli Outboards sono realizzati attraverso le due file di terminali senza saldature presenti sull'SK-10.

Gli Outboards possono essere raggruppati in 7 categorie differenti:

Outboards di display:	LR-6 Lamp monitor Outboard LR-4 Seven-segment display Outboard LR-26 Latch/display Outboard LR-27 Octal latch/display Outboard LR-28 Three digit lach/display Outboard
Outboards di ingresso digitale:	LR-2 Logic switch Outboard LR-7 Dual pulser Outboard LR-5 Clock Outboard LR-20 Monostable Outboard LR-10 Programmable timer Outboard
Outboards di comunicazione:	LR-21 UART Outboard LR-14 TTL/20mA current loop interface Outboard LR-13 TTL/RS-232C interface Outboard
Outboards di funzioni digitali:	LR-19 Lach Outboard LR-22 Decoder Outboard LR-17 Decade counter Outboard LR-18 Binary counter Outboard LR-23 Multiplexer Outboard LR-23 Multiplexer Outboard LR-12 Driver/inverter/NOR Outboard.

Oltre a questi vi è l'LR-1 Power Outboard (Outboard di alimentazione) e l'LR-25 breadboarding station Outboard.

POWER OUTBOARDS

L'alimentazione è applicata alla piastra SK-10 tramite l'*Outboard di alimentazione (Power Outboard)* LR-1 (figura A-2) o l'LR-25 Breadboarding Station Outboard mostrato nella figura A-3.

Figura A-1. Outboard LR-7 visto da sotto. Si notano i due pin di alimentazione e i 4 pin di ingresso-uscita.

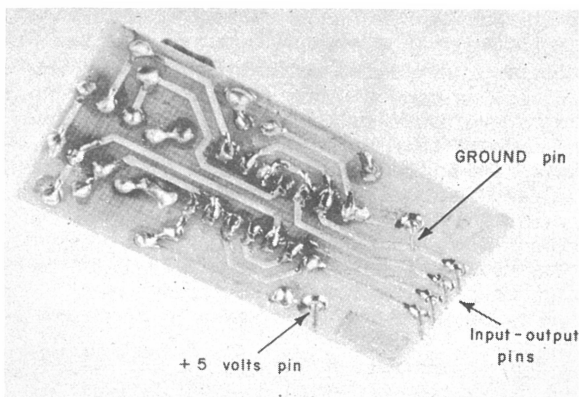



Figura A-2. L'LR-1 Power Outboard (Outboard di alimentazione)

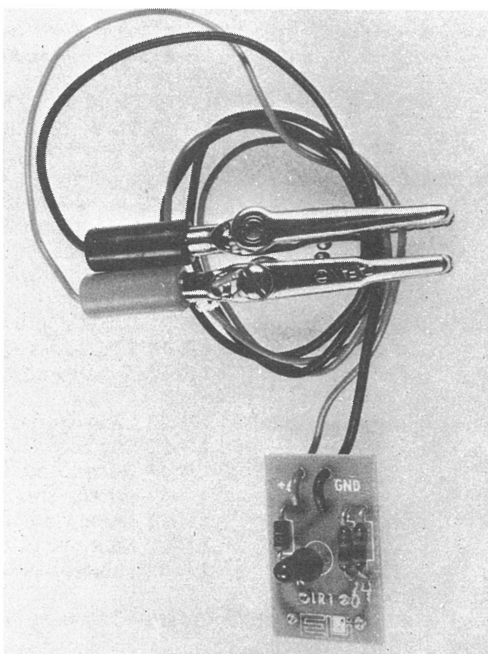



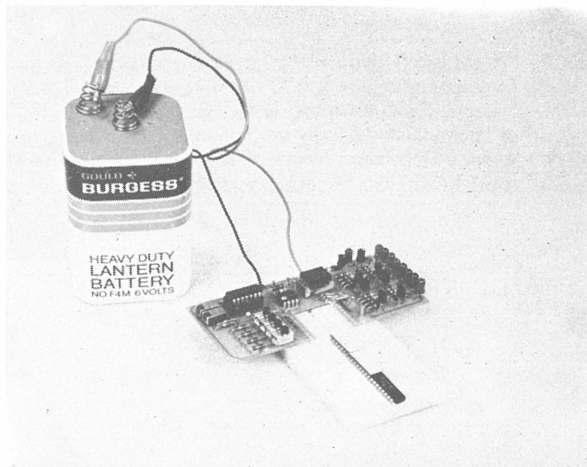
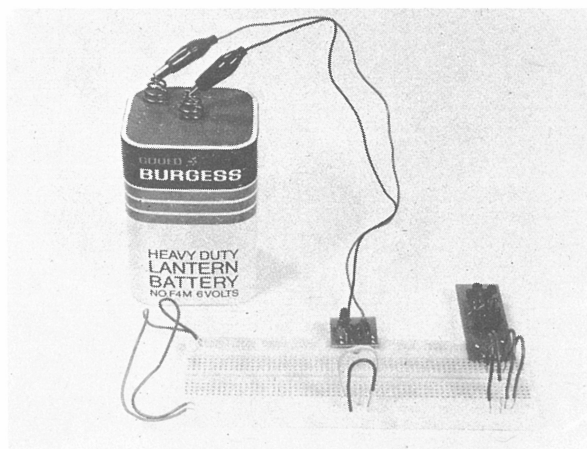
Figura A-3. L'LR-25 Breadboarding Station Outboard montato su una piastra SK-50 e alimentato da una batteria a 6V.

La circuiteria di alimentazione di questi due Outboard controlla la tensione c.c. applicata all'SK-10.
I LED indicano quando è applicata una alimentazione c.c. corretta, quando non è corretta, quando le prese a coccodrillo sono state connesse erroneamente alla batteria o ai terminali dell'alimentatore (i LED sono spenti) e, per ultimo, quando la tensione della batteria è bassa (i LED sono semiaccesi).
Un diodo raddrizzatore previene dannose conseguenze risultanti da errata connessione delle prese a coccodrillo ai terminali della batteria. Se tali prese sono collegate in modo errato, non viene applicata alcuna tensione al breadboard ed i LED restano spenti.
Nella figura A-4 è indicato un esempio di collegamento

Figura A-4. Esempio di utilizzo di una batteria per alimentare l'SK-10. L'LR-1 Power Outboard serve per dare tensione a 50 terminali del bus di alimentazione. I restanti 50 terminali sono collegati ai precedenti con ponticelli, che devono sempre restare collegati.

LOGIC SWITCH OUTBOARDS

Lo *switch logico (logic switch)* è un dispositivo meccanico che presenta ai suoi terminali di uscita uno stato logico 1 o 0.
L'LR- 2 Logic Switch Outboard, indicato nella figura A-5, e anche presente sull'Outboard LR-25 di figura A-3. Fornisce 4 switch logici che possono commutare tra 0 V (0 logico) e +5 V (1 logico).
L'Outboard LR-2 può essere posto indifferentemente sui due lati del breadboard.

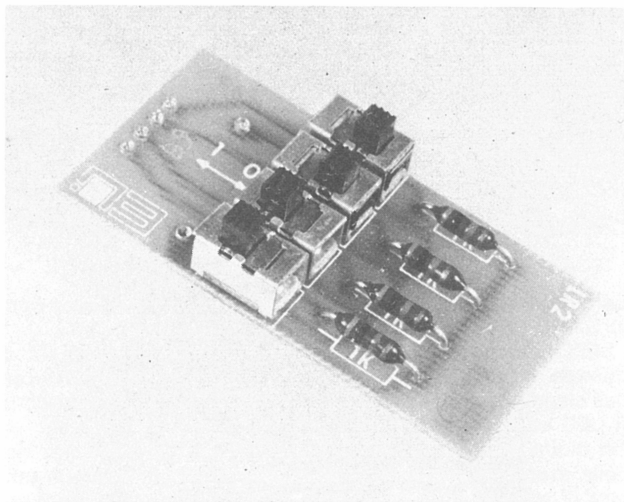



Figura A-5. L'LR-2 Logic Switch Outboard

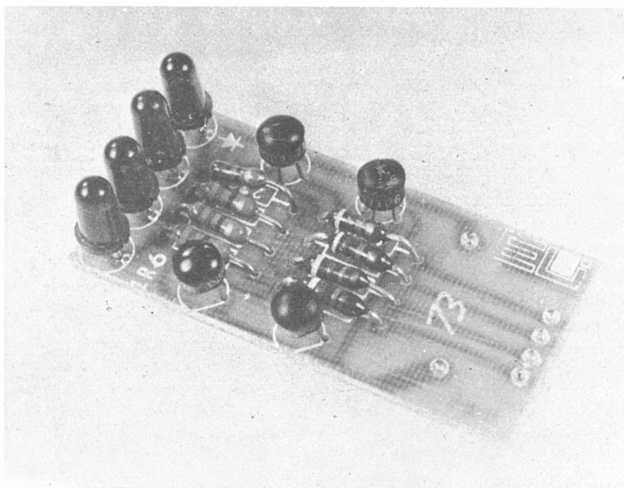



Figura A-6. L'LR-6 LED Lamp Monitor Outboard

LED LAMP MONITOR OUTBOARDS

Un *indicatore a LED (led lamp monitor)* è di fatto un diodo ad emissione di luce (LED) che è acceso nello stato logico 1 e spento nello stato logico 0.
È utilizzato come display per segnali digitali binari.
L'LR-6 LED Lamp Monitor Outboard contiene 4 LED, mentre l'LR-25 ne contiene 8.
L'utilizzo di transistor per pilotare il LED, riduce la corrente per ogni LED a soli 1,5 mA.

PULSER OUTBOARDS

Un *generatore di impulsi (pulser)* è uno switch logico che genera un singolo impulso di clock, che è un ciclo logico completo da 0 logico ad 1 logico ed ancora a 0 logico oppure da 1 a 0 logico ed ancora ad 1 logico.
Per avere un generatore di impulsi, occorre eliminare i *rimbalzi da contatto* causati dagli switch meccanici utilizzati.
Un rimbalzo da contatto (contact bounce), il fermarsi e l'interrompersi di un contatto in modo incontrollato quando i contatti dello switch sono aperti o chiusi, è un fatto abbastanza comune.
In alcune applicazioni digitali, i rimbalzi non sono un problema. In molti circuiti digitali, tuttavia, è necessario che l'uscita di un generatore di impulsi sia esente da rimbalzi, per permetterne l'uso in circuiti digitali *clocked*, come quelli che richiedono un opportuno tempo di clock per le loro operazioni.
L'LR-7 Dual Pulser Outboard, di cui è mostrata una vista dall'alto in figura A-7 e dal basso in figura A-1 utilizza 4 porte NAND a 2 ingressi in un singolo chip di circuito integrato, 7400 per produrre una coppia di *debounced pulsers*, la cui uscita è esente da rimbalzi di contatto. Sono presenti le uscite complementari relative agli stati logici 0 ed 1, dette "0" ed "1", per ogni generatore di impulsi sull'LR-7 ed anche sull'LR-25 (figura A-3).
Per attivare ciascun generatore di impulsi, occorre premere il pulsante di plastica. L'operazione può essere fatta con la punta di una penna a sfera.

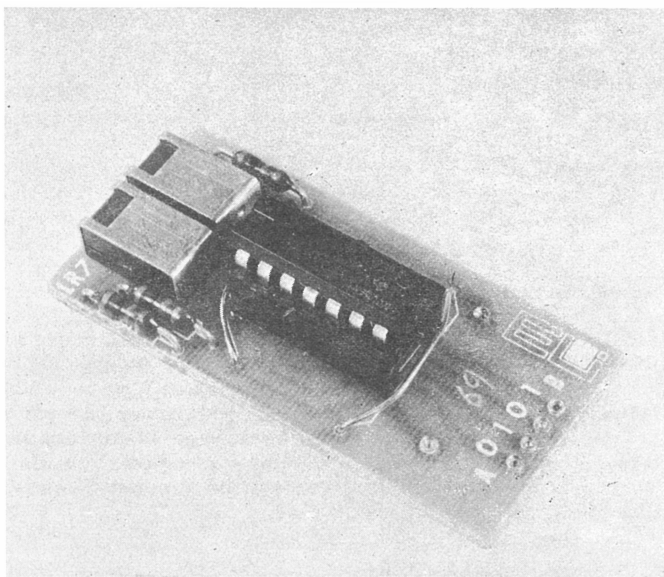



Figura A-7. L'LR-7 Dual Pulser Outboard.

DISPLAY E LATCH/DISPLAY OUTBOARDS

Un *display* è un dispositivo che fornisce la visualizzazione di un segnale elettronico.Un indicatore a LED è un display per un singolo bit di informazione binaria. Quattro LED possono mostrare 4 bit di informazione binaria.

Nell'elettronica digitale, vengono richieste le visualizzazioni delle informazioni:

- Singolo bit
- Gli stati logici di singolo flag, uscite dalle porte, uscite dai flip-flop
- Quattro bit

  Codice binario decimale codificato (BCD), codice esadecimale, ed altri codici binari a 4 bit, come quelli relativi ai microprocessor a 4 bit.

- Codice ottale ad 8 bit

  Codici di istruzione espressi su 8 bit, codici di dispositivo, byte di indirizzo di memoria, byte di dati; tutti utilizzati in microprocessor ad 8 bit come l'MMD-1. Codici ad 8 bit come l'ASCII e l'EBCDIC

- Multipli di 4 bit

  La visualizzazione di numeri decimali in codice BCD nei contatori, misuratori di frequenza, multimetri digitali, pannelli di misura, ed altra strumentazione digitale.

Sono disponibili Outboard per ciascuno di questi tipi di informazioni.

Per visualizzazione di singoli bit, si usa l'LR-6. Per i 4 bit l'LR-6, l'LR-4 Seven-Segment LED Display Outboard (figura A-8) oppure l'LR-26 Latch/Display Outboard (figura A-9). L'Outboard a sette segmenti contiene un display in cui sette segmenti sono posti in modo tale da rappresentare i digit da 0 a 9 attraverso l'accensione selettiva di certi segmenti.

Sull'LR-4 vi è un circuito integrato 7447 decoder/driver, un 5082-7730 della Hewlett-Packard oppure un display numerico Opcoa SLA-1, e sette *resistori limitatori di corrente* per salvaguardare il display.
I quattro ingressi ABCD verso l'Outboard generano i numeri da 0 a 9, cinque simboli ed il blank: quindi in totale 16 diversi stati sono disponibili per il display. Sull'Outboard LR-4 sono presenti 3 ingressi addizionali al chip 7447, il BLANKING INPUT (I), il BLANKING OUTPUT (O) ed il LAMP TEST (T).

L'LR-26 Single Latch/Display Outboard contiene un indicatore numerico singolo od esadecimale della HP. Cinque pin di ingresso all'Outboard funzionano come i 4 ingressi ABCD in codice binario decimale codificato o ingressi esadecimali, più uno strobe (STB) che permette di effettuare un *latch* di 4 bit in ingresso indefinitamente. Tutti gli Outboard di latch/display, l'LR-27, LR-28, LR-26, sono basati sulla serie 5082-7300 di indicatori numerici ed esadecimali della HP. Gli indicatori sono display ad 8 pin, sia decimali che esadecimali, con incorporato un decoder/driver ed un latch. In contrapposizione all'LR-4, i display sono realizzati in termini di matrice a punti 4 x 7, molto comoda da leggere.
Tale matrice permette la rappresentazione di caratteri esadecimali come A, B, C, D, E, F, cioè da numeri decimali da 10 a 15.

Figura A-8. L'LR-4 Seven-Segment LED Display Outboard.
In questo Outboard, la notazione di ingresso 1248 sostituisce la notazione ABCD. La corrispondenza tra i due tipi di notazione è: A=1, B=2, C=4, D=8.

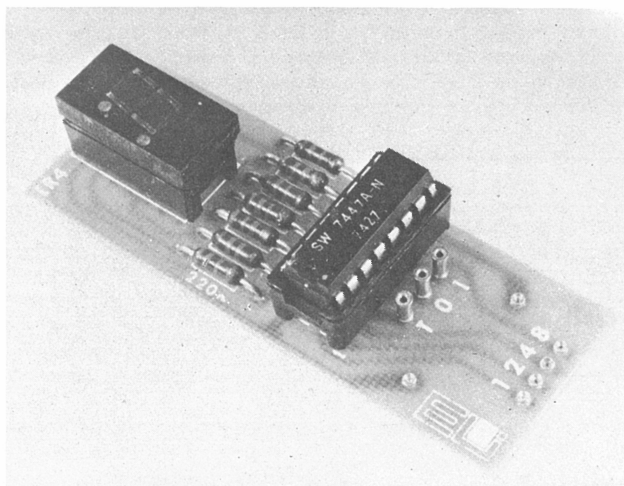



Figura A-9. L'LR-26 Single Latch-Display Outboard

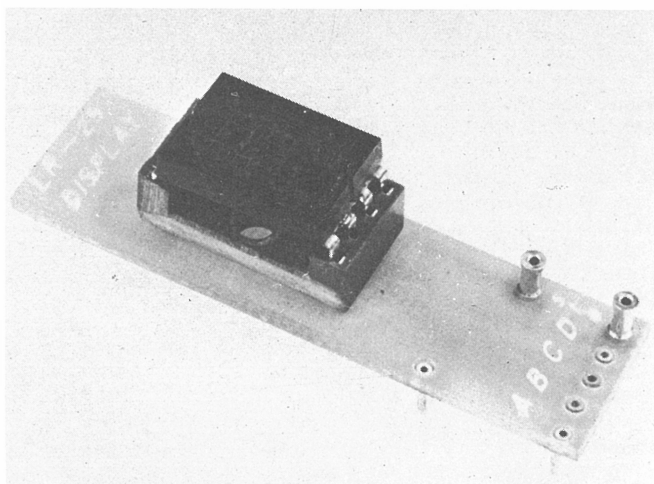



Figura A-10. L'LR-27 Octal Latch/Display Outboard

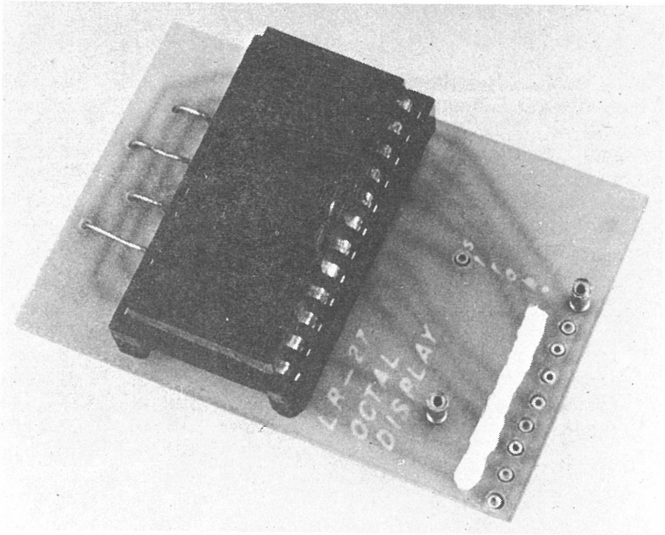
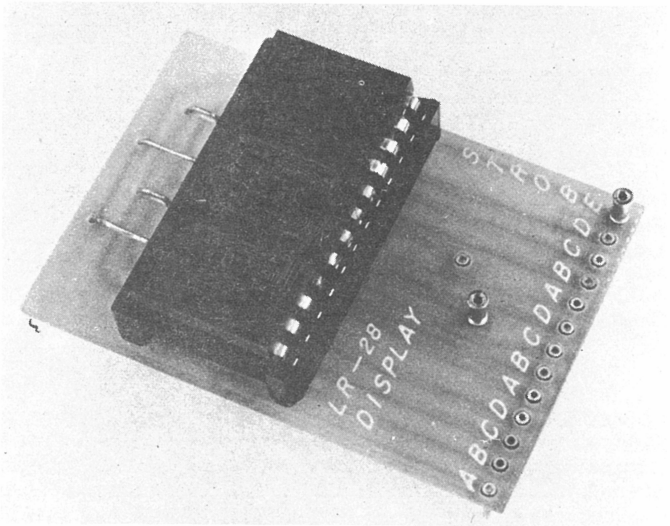



Figura A-11. L'LR-28 Three Digit Latch/Display Outboard

Per la visualizzazione del codice ottale ad 8 bit, discusso nel capitolo 1, si usa l'LR-27 Octal Latch/Display Outboard. Questo Outboard contiene 3 set di indicatori numerici, come si vede in figura A-10, ciascuno dei quali ha l'ingresso D a massa. Due set di indicatori hanno gli ingressi A,B e C, mentre il terzo set ha l'ingresso C a massa e solo A e B come ingressi. Gli ingressi di Strobe (STB) da 3 indicatori sono collegati insieme in modo che risulta possibile memorizzare un'intera parola ad 8 bit espressa su 3 digit. Questo Outboard è utilizzato come *bus monitor* nell'MMD-1 e permette di verificare molti dei trasferimenti dati che avvengono sul bus ad 8 bit del microprocessor 8080A.

Per la visualizzazione di molti digit decimali, ciasuno dei quali è codificato in codice BCD, si usa l'LR-28 Three Digit Latch/Display Outboard. Questo Outboard contiene 3 indicatori numerici od esadecimali e 3 set di ingressi ABCD per ciascun indicatore. Gli ingressi di strobe a tutti e tre gli indicatori sono collegati insieme permettendo così di memorizzare numeri a 3 decimali. Una coppia di indicatori su tale Outboard può essere usata per controllare un byte ad 8 bit di un microprocessor, in termini di due digit esadecimali. Questo Outboard è indicato in figura A-11.

CLOCK OUTBOARDS

Per *clock* si intende un qualunque dispositivo che genera un *impulso di clock,* ciclo logico completo (con questo intendo un segnale che effettua una transizione dallo stato logico 0 allo stato logico 1 e ancora allo stato logico 0 (oppure una transizione dallo stato logico 1 allo stato logico 0 e ancora allo stato logico 1).
L'LR-5 Clock Outboard (figura A-12), presente anche sull'Outboard LR-25 (figura A-3), genera una sequenza di impulsi di clock detta anche *treno di impulsi di clock,* la cui frequenza è funzione della capacità di timing inserita tra due pin visibili sull'Outboard. Tali pin sono facilmente identificabili sull'LR-5 e sull'LR-25. In assenza di condensatori, la frequenza del clock è circa 90 KHz. I condensatori che possono essere impiegati vanno da 10 pF a 100 μF. Nel caso di condensatori elettrolitici, occore identificare il pin negativo; per l'LR-5 è il pin direttamente al di sotto delle lettere LR-5.
Il cuore di questo Outboard di clock è il circuito integrato 555 timer, che presenta una stabilità di frequenza di circa 0,1%. Sull'LR-5, un LED visualizza il corretto funzionamento del clock. Un condensatore di 0,33 μF consente di ottenere una frequenza di circa 0,7 Hz; la vostra frequenza di clock per una capacità identica può mutare del $\pm$ 20% attorno a tale valore. Il pin di uscita dell'Outboard è indicato con CLK. Possono essere aggiunti dei resistori esterni ai restanti ingressi per aumentare il limite superiore di frequenza dell'LR-5. Il pin di uscita dell'LR-25 è indicato con CK, come mostrato in figura A-3.

BREADBOARDING STATION OUTBOARD

L'LR-25 Breadboarding Station Outboard, indicato sia nella figura A-3 che nella figura A-13, comprende i seguenti Outboards: LR-1, LR-2. LR-5, LR-6 (due) LR-7 e LR-26. Questo Outboard viene utilizzato come elemento di breadboarding completo. Con questo Outboard, è possibile attuare qualunque esperimento che richiede non più di 4 switch logici, due generatori di impulsi con logica antirimbalzo, un clock, otto LED, ed un latch/display collegato in parallelo a 4 indicatori a LED;

Figura A-12. L'LR-5 Clock Outboard. La capacità di timing è inserita tra i due pin fra cui vi è la lettera C.

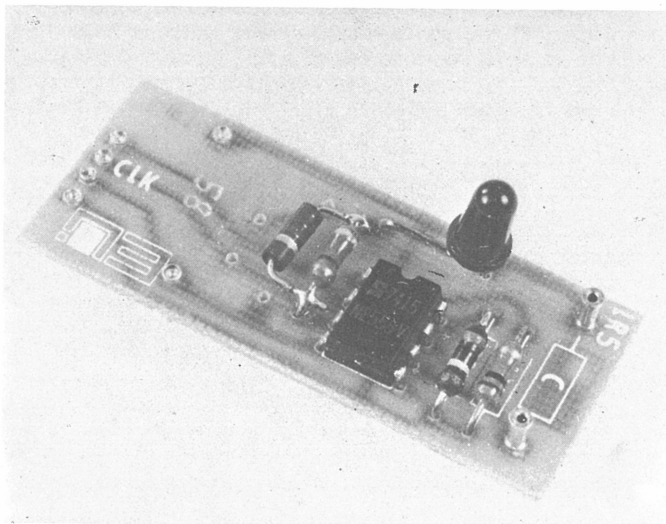
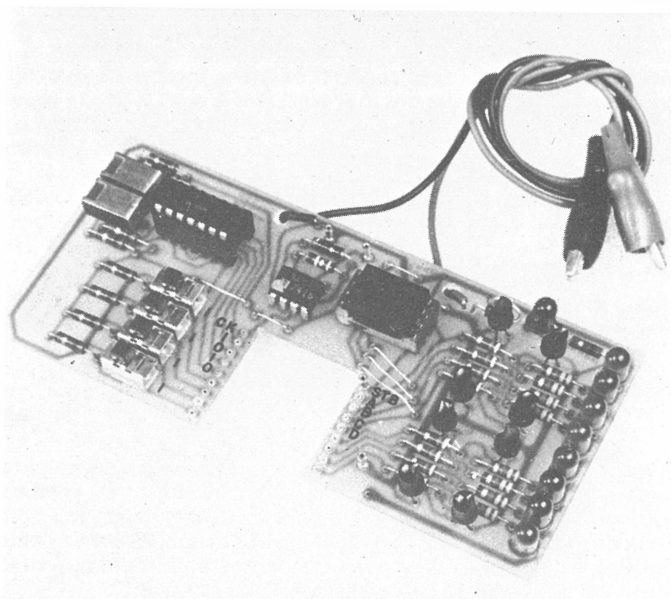



Figura A-13. L'LR-25 Breadboarding Station Outboard.

DECODER OUTBOARD

In uno dei capitoli del Bugbook VI imparerete come decodificare il *"device code" (codice di dispositivo)* di un microprocessor per produrre degli individuali *"device select pulses" (impulsi di selezione di dispositivo)* utilizzabili per abilitare o disabilitare dispositivi digitali elettronici. L'LR-22 Decoder Outboard, mostrato in figura A-14, contiene un 74154, decodificatore singolo chip da 4 linee a 16, e un contenitore DIP a 16 pin. Questo Outboard può essere usato per estendere le 5 uscite di selezione dispositivo presenti sul microcomputer MMD-1 (nel blocco di decodifica I/0). Ogni LR-22 può generare 16 differenti "impulsi di selezione di dispositivo".

MONOSTABLE OUTBOARD

Un *multivibratore monostabile* è un circuito digitale che ha un solo stato stabile, da cui con un impulso di sincronizzazione (trigger) può essere spinto (con un impulso di sincronizzazione-trigger) a cambiare stato, ma solo per un predeterminato intervallo di tempo, al termine del quale ritorna nello stato originario. Tale circuito può essere usato per produrre dei clock singoli di durata definita. L'LR-20 monostable Outboard (figura A-15) contiene un chip multivibratore monostabile retriggerabile 74122 che genera singoli impulsi di clock con l'ausilio di un piccolo potenziometro da 25 kΩ e di una capacità esterna. La durata degli impulsi va da 70 ns a 5 ms.

LATCH OUTBOARD

Un *latch* è un elemento di memorizzazione di una informazione binaria. Un singolo latch memorizza un bit di informazione. L'LR-19 Latch Outboard (figura A-16) contiene un chip 74175 positive-edge-triggered latch. Questo chip è una memoria a 4 bit con ingresso di STROBE o di CLEAR, e due uscite complementari, Q e $\overline{Q}$ per ciascun ingresso. L'Outboard LR-19 è utile per immagazzinare quattro bit di informazione dal microcomputer MMD-1. Un impulso di selezione di dispositivo è applicato all'ingresso di STROBE per acquisire e memorizzare il dato.

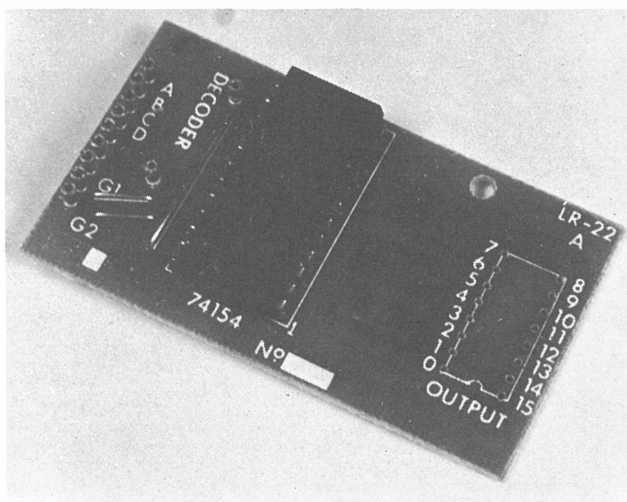



Figura A-14. L'LR-22 Decoder Outboard

Figura A-15. L'LR-20 Monostable Outboard.

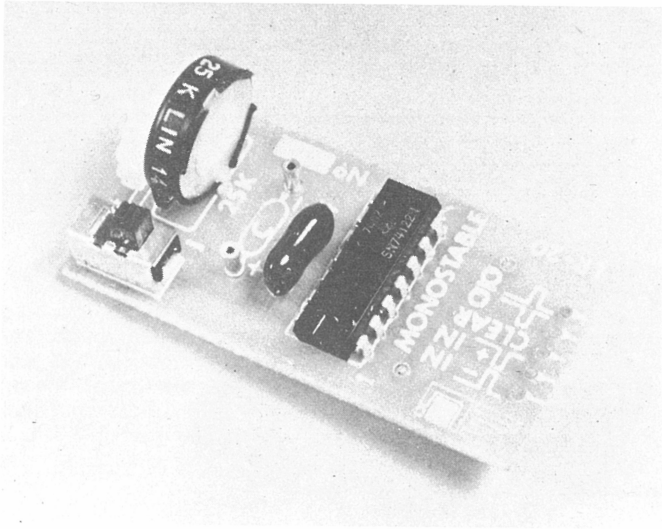
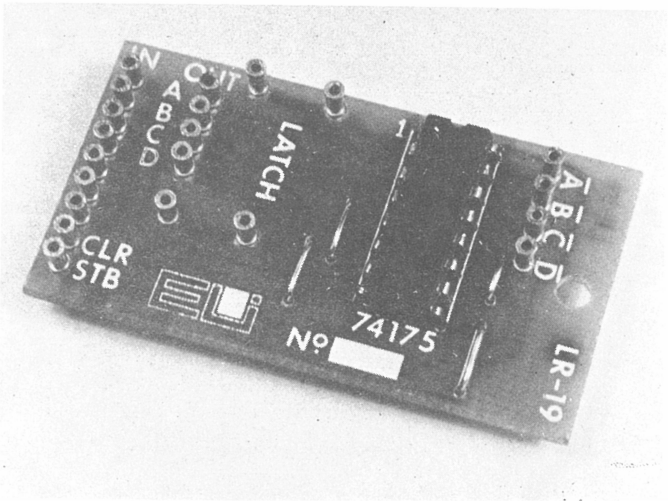



Figura A-16. L'LR-19 Latch Outboard

MULTIPLEXER OUTBOARD

Un *multiplexer* è un dispositivo digitale che può selezionare uno tra più ingressi e trasferire lo stato logico di tale ingresso sull'unica uscita. Questo dispositivo agisce come unidirectional single-pole multiposition switch che passa l'informazione di un ingresso verso un'uscita. L'LR-23 Multiplexer Outboard contiene un chip 74150 da 16 linee ad 1. L'Outboard è mostrato in figura A-17.

COUNTER OUTBOARDS

L'LR-17 Decade Counter e l'LR-18 Binary Counter Outboards sono basati, rispettivamente, sui popolari chip 7490 e 7493.
Un *contatore decimale* (decade counter) è un dispositivo logico che ha dieci stati stabili e può passare sequenzialmente tra questi stati con l'applicazione di dieci impulsi di clock. Un contatore decimale usualmente conta in sequenza binaria da 0 a 9 ricominciando poi ancora da 0. *Un contatore binario* a 4 bit (binary counter) è un dispositivo logico che ha 16 stati stabili e può ciclare tra tali stati con l'applicazione di 16 clock. Ogni contatore, come da figura A-18, contiene uno switch DPDT che permette un conteggio "free running" o che può essere resettato a 0 tramite un segnale applicato a un preciso piedino d'ingresso. Anche l'LR-17 Decade Counter Outboard può essere resettato a 9 attraverso un segnale esterno.

DRIVER/INVERTER/NOR OUTBOARD

Sull'LR-12 Driver/Inverter/NOR/Outboard (figura A-19), un invertitore a *collettore aperto (open collector)* 7405 è utilizzato per generare una porta NOR a due ingressi, due driver e due invertitori. Dato che le uscite sono open collector, possono essere collegate insieme in wired-OR per produrre una porta AND a due ingressi e un addizionale porta NOR a due ingressi.

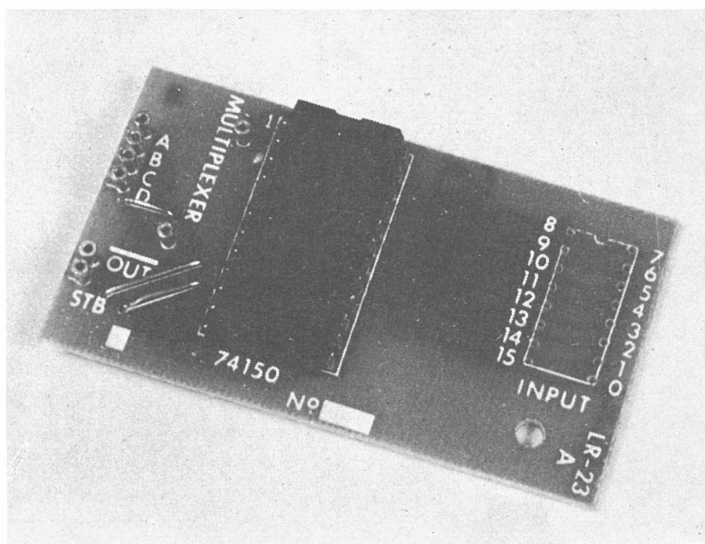



Figura A-17. L'LR-23 Multiplexer Outboard.

Figura A-18. L'LR-17/18 BCD/Binary Counter Outboard. Il tipo di Outboard dipende dal fatto che sia utilizzato il chip 7490 oppure il chip 7493.

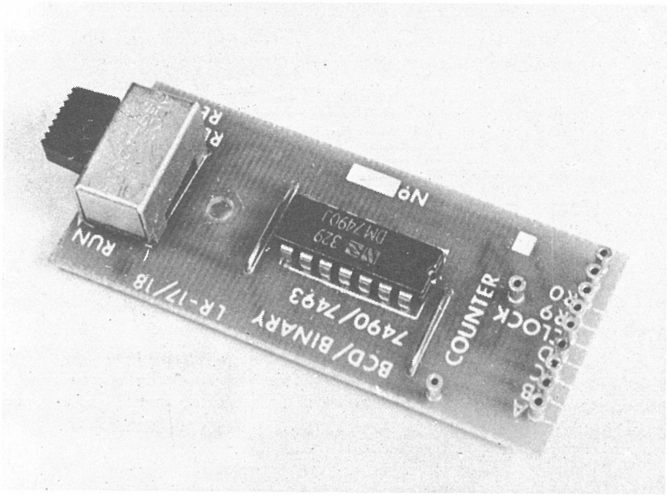
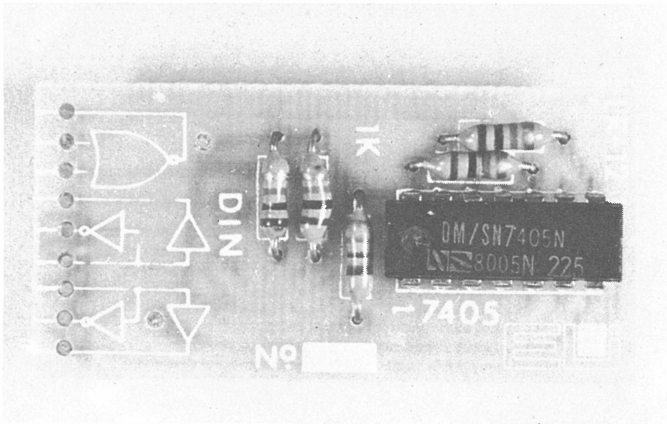



Figura A-19. L'LR-12 Driver/Inverter/NOR Outboard.

UNIVERSAL ASYNCHRONOUS RECEIVER/TRANSMITTER OUTBOARD

Se volete trasmettere dati dall'MMD-1 verso una teletype od un display a raggi catodici, dovete convertire i dati in uscita espressi su 8 bit in codice seriale asincrono ASCII. Come fare ciò è indicato nel Bugbook II, disponibile nella versione Italiana presso la JACKSON Editrice.
L'Outboard utilizzato è l'LR-21 UART, che contiene un chip UART, un "microswitch programming plug" e due contenitori DIP a 16 pin.
L'LR-21 è mostrato nella figura A-20. L'UART può trasmettere dati con una frequenza superiore a 30.000 bits.

TTL/20ma CURRENT LOOP INTERFACE OUTBOARD

Una teletype può trasmettere o ricevere un codice ASCII solo tramite un *"20 mA current Loop"*, descritto nel Bugbook II.
L'LR-14 TTL/20 mA Current Loop Interface Outboard contiene una coppia di GE 4N35 opto-isolatori e un regolatore di corrente di 20 mA on-board. Questo Outboard interfaccia un UART ad un qualunque dispositivo asincrono che opera, relativamente alla trasmissione dati, a frequenze anche superiori a 30.000 bit/s.

TTL/RS-232C INTERFACE OUTBOARD

Desiderando trasmettere o ricevere dati in codice ASCII tramite un *modem* collegato ad una linea telefonica, è necessario convertire i segnali TTL in segnali digitali RS-232C. Questo è possibile con l'LR-13 Line Driver/Receiver e TTL/RS-232C Interface Outboard (figura A-22).
L'Outboard contiene due 8T15 Signetics dual line driver ed un 8T16 dual line receiver, più due diodi zener che assicurano che solo la tensione ±12 V sia applicata al driver.

Figura A-20. L'LR-21 UART Outboard

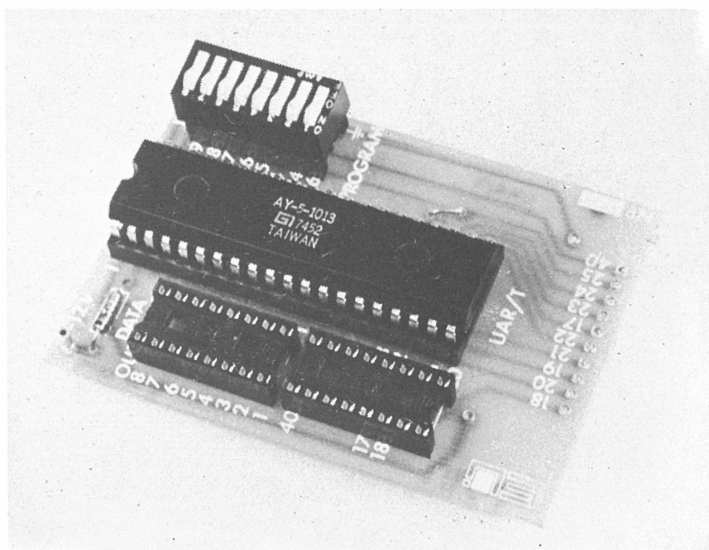



Figura A-21. L'LR-14 TTL/20 mA Current Loop Interface Outboard.

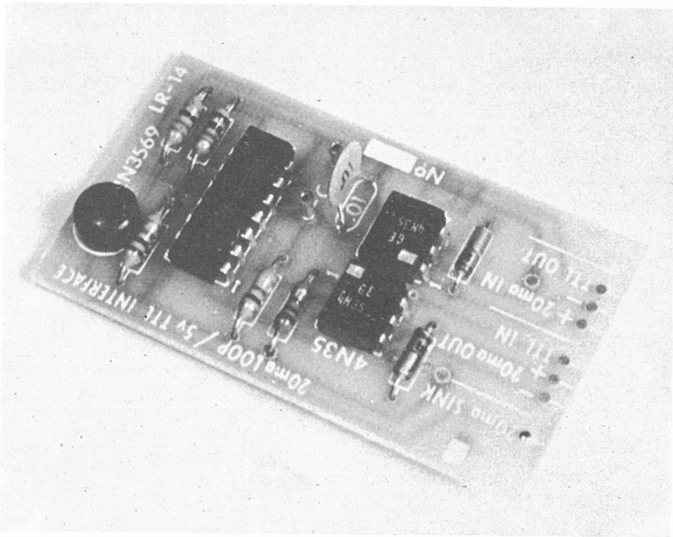
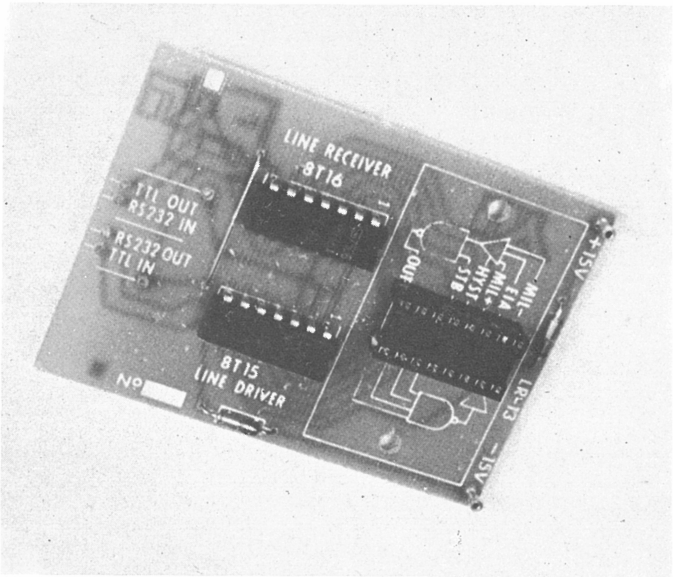
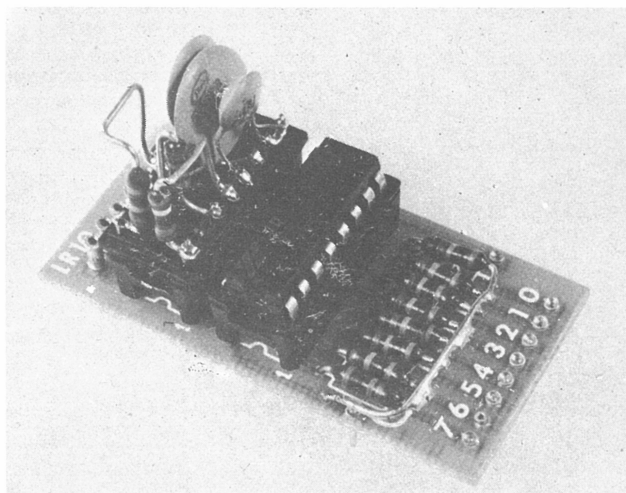



Figura A-22. L'LR-13 Line Driver/Receiver e TTL/RS-232. Interface Outboard

PROGRAMMABLE TIMER OUTBOARD

L'LR-10 Programmable Timer Outboard è basato su un chip XR-2240/2340, timer/counter programmabile, che si presenta in un DIP a 16 pin. Un contenitore programmabile, indicato in figura A-23, programma il chip come timer, circuito di delay, multivibratore monostabile, ecc. Il circuito nella figura è per un semplice clock programmabile, in cui le frequenze sotto riportate possono essere prodotte come multiple della fondamentale, o più bassa, frequenza presente alla uscita del pin 0:

Pin di uscita	Multipli della frequenza fondamentale
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

Si noti che: $2^7=128$, $2^6=64$, $2^5=32$, $2^4=16$, $2^3=8$, $2^2=4$, $2^1=2$, $2^0=1$

Figura A-23. L'LR-10 Programmable Timer Outboard.

APPENDICE 4
TABELLA DI CONVERSIONE FRA CODICI OTTALI E CODICI ESADECIMALI

Ottale	Esadecimale	Ottale	Esadecimale	Ottale	Esadecimale	Ottale	Esadecimale	Ottale	Esadecimale
000	00	063	33	146	66	231	99	314	CC
001	01	064	34	147	67	232	9A	315	CD
002	02	065	35	150	68	233	9B	316	CE
003	03	066	36	151	69	234	9C	317	CF
004	04	067	37	152	6A	235	9D	320	D0
005	05	070	38	153	6B	236	9F	321	D1
006	06	071	39	154	6C	237	9F	322	D2
007	07	072	3A	155	6D	240	A0	323	D3
010	08	073	3B	156	6E	241	A1	324	D4
011	09	074	3C	157	6F	242	A2	325	D5
012	0A	075	3D	160	70	243	A3	326	D6
013	0B	076	3E	161	71	244	A4	327	D7
014	0C	077	3F	162	72	245	A5	330	D8
015	0D	100	40	163	73	246	A6	331	D9
016	0E	101	41	164	74	247	A7	332	DA
017	0F	102	42	165	75	250	A8	333	DB
020	10	103	43	166	76	251	A9	334	DC
021	11	104	44	167	77	252	AA	335	DD
022	12	105	45	170	78	253	AB	336	DE
023	13	106	46	171	79	254	AC	337	DF
024	14	107	47	172	7A	255	AD	340	E0
025	15	110	48	173	7B	256	AE	341	E1
026	16	111	49	174	7C	257	AF	342	E2
027	17	112	4A	175	7D	260	B0	343	E3
030	18	113	4B	176	7E	261	B1	344	E4
031	19	114	4C	177	7F	262	B2	345	E5
032	1A	115	4D	200	80	263	B3	346	E6
033	1B	116	4E	201	81	264	B4	347	E7
034	1C	117	4F	202	82	265	B5	350	E8
035	1D	120	50	203	83	266	B6	351	E9
036	1E	121	51	204	84	267	B7	352	EA
037	1F	122	52	205	85	270	B8	353	EB
040	20	123	53	206	86	271	B9	354	EC
041	21	124	54	207	87	272	BA	355	ED
042	22	125	55	210	88	273	BB	356	EE
043	23	126	56	211	89	274	BC	357	EF
044	24	127	57	212	8A	275	BD	360	F0
045	25	130	58	213	8B	276	BE	361	F1
046	26	131	59	214	8C	277	BF	362	F2
047	27	132	5A	215	8D	300	C0	363	F3
050	28	133	5B	216	8E	301	C1	364	F4
051	29	134	5C	217	8F	302	C2	365	F5
052	2A	135	5D	220	90	303	C3	366	F6
053	2B	136	5E	221	91	304	C4	367	F7
054	2C	137	5F	222	92	305	C5	370	F8
055	2D	140	60	223	93	306	C6	371	F9
056	2E	141	61	224	94	307	C7	372	FA
057	2F	142	62	225	95	310	C8	373	FB
060	30	143	63	226	96	311	C9	374	FC
061	31	144	64	227	97	312	CA	375	FD
062	32	145	65	230	98	313	CB	376	FE
								377	FF

Il **dr. Peter R. Rony** è professore presso il Dipartimento di Ingegneria Chimica del Virginia Polytechnic Institute & State University (USA). L'elettronica digitale ed i microcomputer giocheranno un ruolo molto importante nei controlli di processo, soggetto questo di considerevole interesse per l'ingegneria chimica. Il dr. Rony è coautore di molti altri Bugbooks e di una pubblicazione mensile, denominata Columns, riguardante l'interfacciamento dei microcomputer, che appare su *American Laboratory, Computer Design, Ham Radio Magazine,* in Usa, *Electroniker* in Germania e *Elettronica Oggi* in Italia.

Mr. David G. Larsen è un istruttore del Dipartimento di Ingegneria Chimica del Virginia Polytechnic Institute & State University dove svolge attività didattica a vari livelli nel campo dell'elettronica analogica e digitale. È coautore di altri Bugbook e di una pubblicazione mensile, denominata Columns, riguardante l'interfacciamento dei microcomputer. Con il dr. Rony cura una serie di corsi sui microcomputer, sotto gli auspici della Extension Division della suddetta Università. Questi corsi sono particolarmente apprezzati e seguiti da professionisti di ogni parte del mondo.

Mr. Jonathan A. Titus è presidente della Tychon Inc. La maggior parte della sua attività alla Tychon è concentrata sulle applicazioni di microcomputer e sui piccoli sistemi a microcomputer. Ha scritto ed ha collaborato alla realizzazione di articoli sulla strumentazione e sui microcomputer, pubblicati su riviste sia a carattere professionale che hobbistico. Jon iniziò ad interessarsi ai microcomputer nel '71 quando lavorava all'automazione di strumenti per analisi chimica. La sua prima esperienza fu il microcomputer MARK 8, che era basato sull'8008.. Nel luglio '74 venne presentato dalla rivista Radio Electronics come il primo hobbista di computer. Gli attuali interessi di Jon comprendono lo sviluppo di sitemi didattici, controllo digitale e sistemi di acquisizione dati a distanza.

JACKSON
ITALIANA
EDITRICE

RONY
LARSEN
TITUS

il BUGBOOK V°

ESPERIMENTI INTRODUTTIVI ALL'ELETTRONICA
DIGITALE, ALLA PROGRAMMAZIONE
E ALL'INTERFACCIAMENTO DEL MICROCOMPUTER 8080A

2